Universidad
Carlos III de Madrid

Ingeniería Informática Superior

PROYECTO FIN DE CARRERA

SISTEMA DE MINERÍA DE CORREO ELECTRÓNICO ORIENTADO A REPORTES ESTADÍSTICOS

Autor: Antonio Marcos Viera

Tutor: Francisco Javier García Blas

Leganés, Julio de 2016

Título: SISTEMA DE MINERÍA DE CORREO ELECTRÓNICO ORIENTADO A REPORTES ESTADÍSTICOS

Autor: Antonio Marcos Viera

Director: Francisco Javier García Blas

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 2016 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Resumen

La minería y el análisis de correo electrónico se enmarcan en el análisis de interacciones sociales en el campo de las tecnologías de la comunicación. Basa su estudio en el correo electrónico, una herramienta que envejece con dignidad a la vez que emergen otras redes sociales en las diferentes esferas de la comunicación: personal, académica, laboral y marketing.

Los beneficios que aporta el desarrollo de herramientas y el estudio de grandes corpus de correo electrónico son de aplicación en las tecnologías actuales: detección de *spam*, clasificación de correos electrónicos, análisis y clasificación de contactos, visualización de mensajes. No es menos importante el amplio campo del análisis de las redes sociales donde se estudian las relaciones personales o de liderazgo entre miembros de una organización, búsqueda de los mayores expertos o la detección de comunidades dentro de comunidades.

En este trabajo se plantea una introducción a la minería y el análisis estadístico del correo electrónico que incluye una aplicación para importar el contenido de un gran corpus de mensajes desde varios formatos de archivo: archivos planos de correo, formato de archivo único *mbox*, y formato de archivo PST *de MS Outlook*®; a una base de datos relacional y obtener diversos informes estadísticos.

Palabras clave: Minería de datos, Correos Electrónicos, Enron, Redes Sociales.

Abstract

Email data mining and email analytics belong to social interaction analysis over the communication technologies field. It studies the electronic mail, a tool that has well aged while other social networks arise on different communication spheres such as personal, academic, employment, marketing…

The development of tools oriented to big corpus of email data analysis has several well-known profits such as *Spam* Detection, Email Classification, Contact analysis, Contact classification, and Email visualization. In addition to those, the social network analysis is an important area of investigation where studies exists on Social relations, leadership relations among members of an organization or search of main experts in a net or community detection in the nodes of the social graph.

On this work an introduction to email data mining was done as well as an email analysis application. That system can import big email corpus from three different formats: plan text files, *mbox* unique file format and *MS Outlook*® PST format. The data was imported to a relational database with a custom application to get statistical reports.

Keywords: Email Data Mining, Enron, Social Network

Agradecimientos:

Quisiera agradecer a en primer lugar a mis padres Adolfo y Magdalena por su incansable apoyo, también por su insistencia, para terminar la carrera con este trabajo. Sin ellos no me habría sido posible terminar el presente trabajo.

También quiero agradecer a mi pareja Silvia por su cariño, comprensión y apoyo, ha sido una fuente de inspiración a lo largo de estos años.

A mi antiguo compañero Javier “Escubi”, sin cuyo consejo no habría retomado el Proyecto Fin de Carrera. En la misma línea a mis compañeros de trabajo que también han sido un acicate para mejorar y completar mis competencias algunas de ellas de aplicación directa en este proyecto.

Por supuesto tengo que agradecer también a mi Tutor Francisco Javier por el apoyo y los sabios consejos dados durante la realización del proyecto.

CONTENIDO

Índice de Figuras

Índice de Ecuaciones

Índice de Tablas:

1 INTRODUCCIÓN

1.1 Motivación

El correo electrónico lleva más de tres décadas conformando la espina dorsal del trabajo corporativo. Junto a otras herramientas de comunicación emergentes determina con sus flujos de envíos y reenvíos, puestas en copia, copias ocultas, adjuntos, asuntos e hilos, una semántica propia de relación interpersonal.

El presente trabajo aporta una implementación para el análisis de conjuntos voluminosos de correo electrónico, pero sobre todo una metodología para este tipo de proyectos de análisis, sin dejar de lado capacidades de extensión para abordar nuevos problemas y análisis.

Como caso de estudio se toma el corpus de correos de la corporación Enron que ha sido fuente de datos para muchos otros estudios realizados con objetivos similares: **[Klimt]**, **[SNAP]**, **[Tang]** o **[Vadher]** entre otros.

Enron Corporation era una compañía americana con sede en Houston que antes de su quiebra a finales del 2001 contaba aproximadamente con 22.000 empleados. Llegó a alcanzar el lugar de una de las primeras empresas del mundo en el sector de la electricidad y el gas natural (entre otras actividades) y tenía un valor de negocio cercano a 101 billones de dólares en el año 2000.

La quiebra de Enron ocurrió a la sombra de un fraude contable sistemático sin parangón hasta la fecha. La comisión federal de regulación de la energía hizo público el conjunto de datos y se colgó en internet en el transcurso de la investigación. Según se cita en **[Vadher]** los datos proceden de los ficheros de *log* del servidor POP de correo. Estos archivos contienen la mayor parte de los correos electrónicos de 150 empleados de la empresa durante el periodo 1998-2002.

Desde entonces, el corpus de correos electrónicos de Enron ha sido objeto de cuantiosos análisis y estudios en diversas áreas tanto por la comunidad científica como por el sector privado.

En todos los estudios de minería y análisis de correo electrónico, se requiere conocer ciertos datos estadísticos del conjunto de datos, ya bien para guiar los pasos de análisis siguientes ya bien para contrastar las hipótesis.

1.2 Objetivos

El objetivo principal del presente proyecto es formalizar un método de análisis de grandes grupos de datos de mensajes de correo electrónico. Para la sistematización de dicho proceso se ha de tener en cuenta el estado de avance en el campo de la minería del correo electrónico, en el que se ha de profundizar para disponer de un método de análisis.

También es objetivo, no menos importante, la implementación de un sistema de importación para corpus de correos electrónicos a una base de datos relacional y la obtención de informes de carácter estadístico sobre los mismos.

Otros objetivos secundarios de menor calado son:

- El desarrollo y despliegue de un sistema basado en tecnologías de *Microsoft* que ejemplifica el desarrollo del anteriormente citado método de análisis.
- El resumen y análisis de los principales elementos de las gramáticas formales que definen las principales especificaciones de mensajes de correo electrónico.
- La formulación de estrategias de abordaje para realizar un análisis de un conjunto de mensajes de gran volumen.

1.3 Estructura del documento

Esta memoria se estructura en los siguientes capítulos:

2. Estado de la cuestión

Se presenta una visión panorámica de las principales finalidades en las que se utiliza la minería de correo electrónico así como las técnicas aplicadas en diversos escenarios. Se incluye una relación de aplicaciones y sistemas relacionados y se concluye con los conjuntos de datos públicos más relevantes.

3. Estándares y protocolos del correo electrónico

Se presenta un resumen de los protocolos y estándares principales que son de mayor interés para realizar un análisis con éxito. En la segunda parte del capítulo se aborda el eje principal de la gramática formal que describe la estructura sintáctica de los mensajes.

4. Estrategias de abordaje

Se reproduce la estrategia empleada para abordar el estudio y análisis de un gran corpus de correos electrónicos. Cada fase del proceso puede tener sus particularidades así como sus variantes en función del objetivo concreto de cada estudio.

5. Análisis y Diseño

Se expone el análisis y el diseño del sistema de importación y análisis sobre el que versa el presente proyecto fin de carrera. En la primera parte del capítulo se presentarán los casos de uso, una descripción detallada y los requisitos de usuario de capacidad y restricción. En la segunda parte, que trata del diseño hay una enumeración de los componentes y módulos del sistema así como el diseño

de la base de datos.

6. Detalles de Implementación

Se detallan algunas cuestiones que han surgido durante la implementación y se ha considerado que son remarcables como la *API* de *Java Mail*, el empleo de funciones *CLR* en la base de datos para disponer de mejores expresiones regulares y una breve introducción al uso de *Reporting Services*.

7. Validación

En este capítulo se organiza el plan de pruebas y se presenta la matriz de trazabilidad que relaciona los requisitos del capítulo de análisis y diseño con las pruebas del plan de prueba.

Se analiza el rendimiento del sistema, su extensibilidad para dar cabida a nuevos estudios y análisis.

8. Conclusiones y líneas futuras

En este capítulo se sacan conclusiones a la luz de los resultados alcanzados y se trazan posibles líneas de estudio futuro. También se aborda la metodología empleada, un presupuesto para desplegar e implementar el sistema y la planificación del trabajo realizado.

1.4 Definiciones y acrónimos

- **BD (BBDD)**: acrónimo de Base de datos (bases de datos en plural).
- **CLR**: Siglas de *Common Language Runtime*, es un entorno de ejecución donde corre el código *.NET* y que provee servicios que simplifican el proceso de desarrollo.
- **ETL**: Siglas de *Extract*, *Transform and Load*, es el término utilizado para referirse a los procesos empleados para mover datos desde diversas fuentes, limpiarlos y cargarlos en una base de datos, *Data Mart* o *Data Warehouse* para analizar o apoyar procesos de negocio.
- **IP**: *IP Address*, o dirección de internet, es una etiqueta numérica asignada a cada dispositivo que participa en una red de computadoras que utilizan el protocolo de internet (*Internet Protocol*), se emplea para identificar y direccionar los mensajes.
- **Q1, Q2, Q3, Q4**: Es el acrónimo anglosajón (*Quarter*) para el primer, segundo, tercer y cuarto trimestre del año respectivamente. Según el contexto puede referirse a trimestre fiscal o natural.
- **NET, .NET**: *Framework* de *Microsoft* que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que proporciona herramientas de desarrollo

rápido de aplicaciones.

- ***Phising*:** actividad delictiva que implica una suplantación de identidad, habitualmente de un proveedor de servicios como la banca online, frente a un usuario particular para obtener la clave y usuario de éste.
- **RFC**: acrónimo de *Request for Comments*, Petición de Comentarios; es el nombre que se da a los documentos presentados por grupos expertos en distintas áreas para la estandarización a nivel mundial de protocolos.
- ***Spam***: Es el nombre que se da a los correos electrónicos no solicitados por el usuario que generalmente contienen publicidad no deseada enviados en grandes cantidades que perjudican al receptor del mismo. El nombre proviene de un show humorístico de los *Monthy Pyton.*
- **TSQL**: siglas de *Transact SQL*, el lenguaje propietario de consultas *SQL* del sistema gestor de bases de datos *MS SQL Server*.
- **URL**: siglas de *Uniform resource locator*, (localizador de recursos uniforme), es un identificador de recursos uniforme (*URI*) cuyos recursos referidos pueden cambiar, esto es, la dirección puede apuntar a recursos variables en el tiempo.

2 ESTADO DE LA CUESTIÓN

En este capítulo presentaremos una panorámica de los objetivos, métodos, herramientas de análisis y las fuentes de conjuntos de datos de correo electrónico.

A la publicación de los correos de Enron han seguido un buen número de estudios en materia de redes sociales y correo electrónico. Muchas empresas han utilizado esos datos para exhibir el funcionamiento de sus aplicaciones, construir bancos de pruebas para algoritmos y realizar entrenamiento de programas. Una lista de las distintas finalidades del análisis y minería de correo electrónico más populares sería: la detección de *spam*, la clasificación de correos electrónicos, el análisis de los contactos, las propiedades del correo electrónico como red social y otras tantas que han surgido y que sin dudas surgirán en el futuro.

2.1 Finalidades de la minería de correo electrónico

La minería de correo electrónico ha sido estudiada en la última década (cf.: **[Tang]**) para abordar cinco áreas principales: detección de *spam*, clasificación de correos, análisis de contactos, análisis de las propiedades como red social y ayuda a la visualización.

2.1.1 Detección de *Spam*:

Es la principal área en el que se emplea la minería de datos sobre correos electrónicos. Según **[Trend Q1-2010]**, las estimaciones del coste que suponen estos correos no deseados ponen de manifiesto la necesidad de entrenar y explorar sistemas clasificadores. Este mismo informe apuntaba que en 2010 el *spam* suponía el 83% del tráfico de correos en el primer trimestre de ese año y la tendencia desde entonces ha sido al alza.

En éste área se utiliza una serie de métodos denominados de clasificación o clasificadores. Los clasificadores se pueden considerar funciones que deben dividir los mensajes en las categorías de *spam* y no *spam* con un margen de error para falsos positivos menor de 1%: la probabilidad de clasificar como *spam* un correo legítimo es muy baja. Existen numerosas aproximaciones dentro de estos métodos que hacen que se clasifiquen en dos familias: basados en el contenido y basados en el emisor **[Tang]**.

Clasificadores de *Spam* basados en el contenido:

Determinan si un correo es *spam* en base a su contenido, se requiere de corpus de mensajes para entrenar o construir los clasificadores.

- La clasificación se basa en el desarrollo de clasificadores que tratan cada mensaje como un vector, los más populares son los clasificadores de **Naïve Bayes**, ***Support Vector Machines* (SVM)** y los basados en reglas.
- El agrupamiento *semi-supervisado* (***Semi-Supervised Clustering***) es otra modalidad basada en el contenido donde se definen grupos de mensajes que son etiquetados como *spam* o no *spam*, con un elemento central en torno al cual se encuentran otros correos similares. A veces se utiliza un umbral de distancia, para ayudar a obtener distintas sensibilidades.

Clasificadores de *Spam* basados en el emisor:

Como su nombre indica utilizan propiedades del emisor para determinar si un mensaje, tratado también como un vector, es o no *spam*. Los métodos más comunes son también la clasificación, la agrupación *semi - supervisada* y el análisis de la reputación del emisor.

Siguen la orientación vista para los métodos de detección basados en el contenido con algunas diferencias: en la clasificación se emplea el estilo lingüístico y el nombre del emisor.

El análisis de reputación se basa en establecer una valoración de un emisor inferida por una red de reputaciones o por su dirección IP, cuantos más correos de *spam* envía un emisor peor es su ratio y más probabilidades hay de que el correo que envía sea *spam*.

2.1.2 Clasificación de correos electrónicos

Se trata de catalogar los correos en clases que tenga sentido para el usuario del correo. Formalmente se define como la asignación de los correos a alguna categoría en función de determinadas características. Algunas veces se ve la detección de *spam* como un caso particular de la clasificación de correos en una categorización binaria: *spam* y *no spam*.

La necesidad para categorizarlos surge a partir de estudios que afirman que la clasificación manual del correo consume, según se cita en **[Bellotti]**, hasta un 10% del tiempo que los usuarios destinan a gestionar su correo electrónico.

Como es un problema similar a la detección de *spam*, se aplican algunos de los métodos vistos para ese caso:

- **Naïve Bayes**
- *Support Vector Machines* (**SVM**),
- Clasificadores **TF-IDF**: es el método más común. Consiste en calcular la distancia entre el correo (recordemos que todos estos métodos tratan a cada mensaje como un vector) y cada una de las categorías y asignarlo a la categoría más próxima.

El éxito de estos métodos de clasificación se suele contrastar con encuestas y entrevistas a usuarios. De estas últimas se obtienen directrices para mejorar los métodos empleados.

2.1.3 Análisis de contactos

Se define como contactos a los emisores y receptores de mensajes en el corpus. Las técnicas de análisis de contactos tratan de identificar y agrupar los contactos mediante el estudio de características concretas en el corpus y en aspectos de la red de comunicación. Uno de los objetivos en este área es la construcción de aplicaciones que dado un conjunto de datos permitan encontrar a los expertos en un campo de conocimiento concreto. El proceso se suele dividir en dos fases: por una parte la identificación de contactos y por otra la clasificación de los mismos.

Un buen número de los métodos presentan como mayor hándicap que encuentran los expertos percibidos pero no los expertos reales. La brevedad en el contenido de los correos puede suponer otra

dificultad, si los correos son demasiado cortos no se pueden extraer marcas representativas para detectar los alias.

Identificación de Contactos

Se trata de encontrar contactos con características especiales. Se aplican con varios propósitos, por ejemplo pueden ayudar a encontrar especialistas en determinadas áreas dentro de una organización grande, o distinguir cuando el emisor de un mensaje sospechoso es el verdadero dueño de esa cuenta u otra persona con malas intenciones.

En la identificación de contactos se utilizan métodos similares a los mencionados clasificadores, **SVM**, **TF-IDF**, **K-NN** (K-*Nearest Neighbours*) y métodos de análisis de redes sociales **SNA** (*Social Network Analysis*).

Clasificación de contactos

Consiste en agrupar los contactos de correo electrónico de tal modo que los de cada grupo compartan ciertas características. Un ejemplo típico es utilizar los contactos de un grupo como receptores sugeridos cuando se está escribiendo un mensaje una vez se ha seleccionado uno o más contactos como receptores.

Las técnicas más populares para esta clasificación suele ser intercambiar frecuencias y contenidos. Se usan también variantes de los métodos de análisis social **SNA** (*Social Network Analisys*) y de agrupación y también el algoritmo de **Girvan-Newman** (véase **[Girvan y Newman]**).

Otros estudios al respecto

En **[Klimt]** se proporciona una breve introducción al conjunto de datos de Enron para la labor de clasificar los contactos. No es se extiende más allá de dos páginas pero es la línea de salida más utilizada para abordar el estudio de los correos de Enron.

El objetivo del estudio es analizar la capacidad que ofrece el conjunto de datos para explorar el modo en el que los humanos organizamos nuestros mensajes de correo en carpetas.

El estudio concluye que el corpus de correo electrónico de Enron es un conjunto aceptable para analizar los hábitos de clasificación humana de correos en carpetas y además apunta a que es válido para realizar análisis sobre hilos.

2.1.4 Análisis de las propiedades como red social

Un corpus de correo electrónico puede verse como una red de nodos: direcciones de correo, o

personas detrás una o más direcciones, y aristas entre los nodos que representan el envío de correo. Según la naturaleza del estudio y el propio algoritmo varían las propiedades de las aristas: dirigidas, sin dirigir, con pesos, etc.

Se suelen contemplar dos tipos de redes, unas son las llamadas **egocéntricas** o locales a un usuario, que contienen sus mensajes de correo y donde pueden aparecer más comunicaciones entre otras personas: listas de distribución e hilos de conversación, y las redes completas de una organización en las que se disponen de los correos electrónicos de un buen número de personas.

Las redes sociales del correo electrónico no son las únicas que se han estudiado en este ámbito pero son significativas por almacenar una información muy rica de la comunicación entre personas y por ello no solo se estudian para mejorar las herramientas software de correo electrónico, sino también para entender las relaciones humanas en comunidades virtuales.

Se emplean aquí métodos de análisis social de redes **SNA** y métodos de clasificación.

Métodos de análisis de redes sociales

Son los más utilizados, analizan las propiedades de la red desde el punto de vista estadístico para proporcionar información sobre las distintas dimensiones de estudio dentro la red. Se basan de manera intensiva en la teoría de grafos, conceptos como el grado de entrada y salida de un nodo, caminos más cortos, etc…, son los que se utilizan para obtener valores estimados de importancia social. La *intercesión* de un nodo ***v*** (***betweenness***) se mide como la suma de la proporción de las comunicaciones entre otro nodo ***i*** y otro nodo ***j*** que pasan por ***v*** dividido por las comunicaciones que hay directamente entre ***i*** y ***j***.

$$BW(v) = \sum \frac{G_{ivj}}{G_{ij}},$$

Ecuación 1: Fórmula de la intercesión (betweenness).

Esta ***intercesión*** es un concepto clave para determinar la importancia social de un nodo, pero también se estudian otros como el tiempo de respuesta al contestar un mensaje recibido y la probabilidad de respuesta.

Métodos de clasificación

Se pueden utilizar como una herramienta para obtener o predecir información del contexto social a nivel individual (cuenta de correo o persona).

Se ha aplicado **SVM** al análisis de redes sociales, tomando como categorías la naturaleza del correo electrónico: informar, preguntar y planificar. Como ejemplo de características analizadas se pueden

citar los signos de puntuación finales, URL y la frecuencia de uso de ciertos *emoticonos*.

Estudios al respecto

En **[Carn]**, tenemos un estudio bastante amplio que aborda el problema de definir comunidades sobre grafos muy grandes donde cada nodo representa una persona, o una entidad, y cada arista es una vía de comunicación.

Recoge algunas de las teorías comunes y añade particularidades técnicas en el método de valoración de la bondad de las comunidades que emergen con ese tipo de algoritmos. Compara muchos grafos computados a partir de conjuntos de datos de tamaños heterogéneos como los correos de Enron.

2.1.5 Visualización de correos

En líneas generales las técnicas de visualización del correo electrónico tienen la finalidad de ayudar a los usuarios de correo para identificar, recuperar y resumir información útil, oculta bajo el gran volumen de mensajes. Los estudios existentes en visualización se centran en identificar problemas de interacción con el correo electrónico de clientes de correo y productos ya existentes. Las mejoras en los mismos derivadas de estas técnicas se miden siempre por encuestas y entrevistas a usuarios.

2.1.6 Otras Finalidades

Recomponer hilos por similitud.

Existe un estudio interesante al respecto que propone una estrategia para recomponer hilos de comunicación a partir del texto que se reenvía como conversación anterior en respuesta a un mensaje **[Yuan y Harnly]**. Generalmente este texto está separado por varios tipos de marcas identificativas, que son las que la mayoría de clientes de correo utilizan para sombrear u ocultar la parte de conversación anterior. Analizan cada caso de estas marcas para poder inferir y reconstruir los hilos de correo.

El contraste que se hace para validar el método es estadístico, se revisan un número de correos y se extrapola el porcentaje de éxitos al resto de mensajes del conjunto.

2.2 Métodos de análisis

En esta sección vamos a exponer de modo resumido la filosofía subyacente de algunos de los métodos de análisis citados anteriormente.

2.2.1 Naïve Bayes

Es el nombre que se da a una familia de clasificadores probabilísticos que se basan en el teorema de Bayes bajo el supuesto de una fuerte independencia entre las características. Este tipo de clasificadores también se utilizan en la categorización de textos.

Se parte de una serie de características organizadas en un vector *x* = (x_1,… x_n) donde x_i es una característica que no guarda correlación significativa con el resto de características. Con un conjunto de correos de entrenamiento se asigna probabilidades a que ocurra un determinado vector en el conjunto, tal que ese vector pertenezca a una clase C_k. En el caso de la detección de *spam* tendríamos dos clases C_1 y C_2 por ejemplo. En el problema de la clasificación de mensajes de correo electrónico podríamos, en un escenario simplificado, determinar las clases por inspección o por un término muy frecuente en el asunto o en el cuerpo y de ese modo tendríamos C_1, C_2… C_n clases.

Utilizando el teorema de *Bayes* partiríamos de la fórmula:

$$p(C_k|\mathbf{x}) = \frac{p(C_k)\ p(\mathbf{x}|C_k)}{p(\mathbf{x})}.$$

Ecuación 2: Teorema de Bayes aplicado a la probabilidad de una clase.

Utilizando la regla de encadenamiento para aplicaciones repetidas de la definición de probabilidad condicional llegaríamos a la fórmula:

$$p(C_k|x_1,\dots,x_n) = \frac{1}{Z}p(C_k)\prod_{i=1}^{n} p(x_i|C_k)$$

Ecuación 3: Teorema de Bayes desarrollado.

Donde ***Z*** sería la evidencia: p(x), es decir la probabilidad de que se dé un vector concreto. Por tanto ***Z*** es un valor de escalado que depende de las características x_1,…x_n. Si la probabilidad de cada característica es conocida el valor ***Z*** es constante.

El resto de la fórmula nos diría que la probabilidad de que un correo electrónico sea de la clase C_k multiplicado por el *productorio* de la probabilidad de que una característica *i* esté condicionada a que el correo pertenezca a la clase en cuestión.

Hay otros modos de construir clasificadores dentro de la Familia Naive-Bayes los más populares son el ***Multinomial Naive Bayes*** y ***Bernoulli Naive Bayes***.

$$p(\mathbf{x}|C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{x_i}$$

Ecuación 4: Clasificador Multinomial (Naive-Bayes).

$$p(\mathbf{x}|C_k) = \prod_{i=1}^{n} p_{ki}^{x_i}(1 - p_{ki})^{(1-x_i)}$$

Ecuación 5: Clasificador de Bernoulli (Naive-Bayes).

2.2.2 SVM (*Suport Vector Machines*)

Las máquinas de vectores de soporte, ideadas por *Vladimir Vopnik* y su equipo en AT&T se basan en algoritmos de aprendizaje supervisado. Están muy próximas a las redes de neuronas **[Isasi]**, siendo máquinas que hay que entrenar con un conjunto de datos.

Como en otros tantos métodos de clasificación supervisada se entrena a la máquina con un conjunto de vectores, donde existen variables atributo que actúan como variable predictora y la *característica* que es un atributo empleado para definir un híper-plano que separe los vectores posibles en dos categorías. *SVM* busca los planos que diferencian las categorías dejando el mayor margen posible entre sus elementos.

Por ejemplo en la siguiente figura:

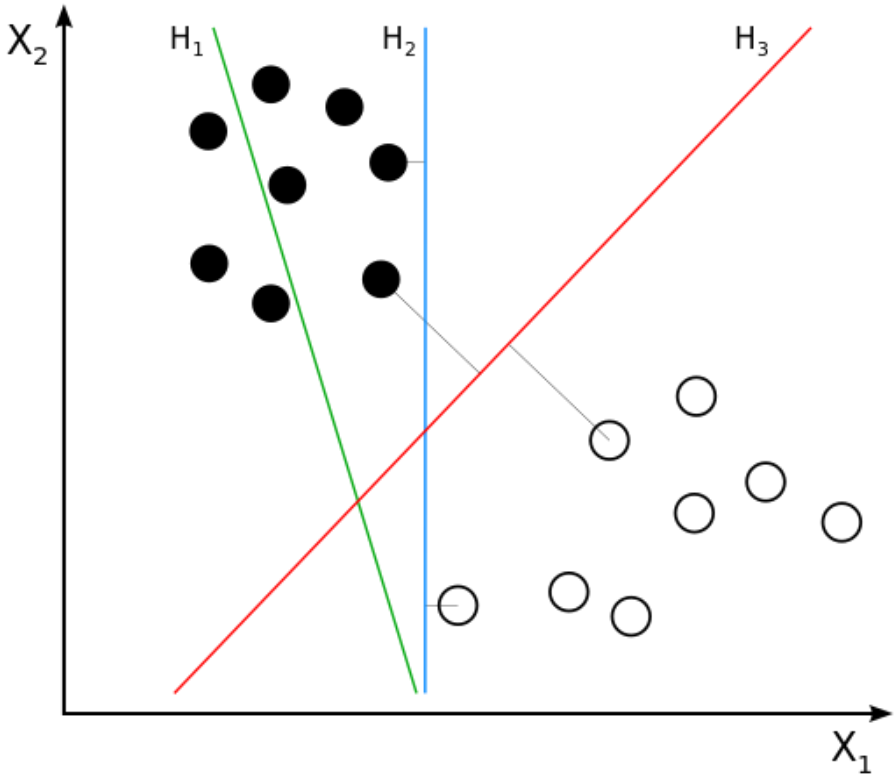


Figura 1: Ejemplo de planos clasificadores (Fuente Wikipedia).

Se presenta un conjunto de vectores bidimensionales, y tres rectas que hacen la función de híper-plano (en casos más complejos las funciones que definen los híper-planos son mucho más complicadas).

En el ejemplo H_1 no separa a los vectores en dos categorías, H_2 los separa pero no consigue el mayor margen de distancia entre los dos grupos mientras que H_3 si consigue dividir bien los vectores. Como el objetivo del entrenamiento es tener una máquina que clasifique bien los vectores reales es importante que el híper-plano generado separe bien los conjuntos de prueba.

Como sucede con las técnicas de redes neuronales, pueden tener problemas de sobre-entrenamiento, en especial si el conjunto de entrenamiento no es suficientemente representativo de los datos reales.

Esta familia de métodos contiene una variante para clasificar en más de dos categorías. En el ejemplo propuesto se separan las clases con una función lineal pero pueden darse híper-planos más complejos en problemas reales para los que se utilizan funciones *Kernell*:

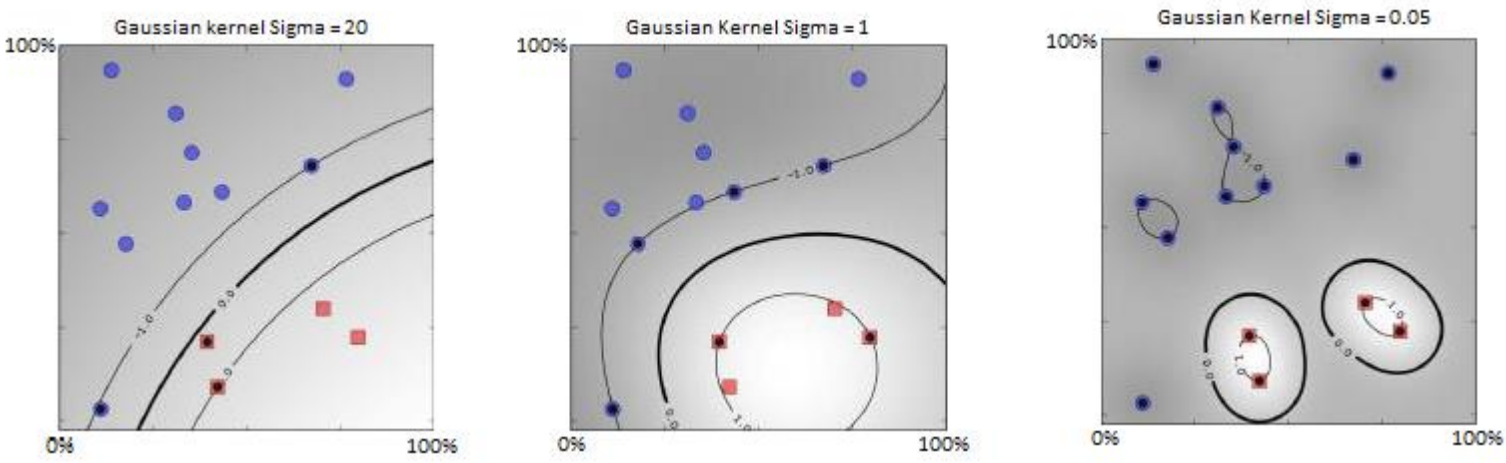


Figura 2: Ejemplos más complejos de clasificación (fuente Wikipedia).

2.2.3 TF-IDF

Su nombre es acrónimo del término "*Term frequency – Inverse document frequency*", en la literatura castellana se traduce como "Frecuencia de término - frecuencia inversa de documento". Constituye un método de valoración para la relevancia de un término, una palabra, dentro de un documento en el contexto de una colección de documentos. Se utiliza con frecuencia como factor de ponderación en la búsqueda y recuperación de información y en la minería de texto.

La medida más obvia para un término es contar el número de veces que aparece esa palabra en el texto, pero esta valoración puede estar contaminada si es un término común del lenguaje. Ahí es donde entra en juego la colección de documentos que requiere el algoritmo: la relevancia del término se compara con la relevancia del mismo en toda la colección de documentos, así términos comunes como artículos y preposiciones, o sustantivos muy presentes en el dominio del problema no se consideran importantes si su relevancia en el documento no es muy superior a la que tiene en la colección entera de documentos.

En resumen, es un valor que aumenta proporcionalmente al número de veces que una palabra aparece en el documento, pero compensada por la frecuencia de la palabra en la colección de documentos.

La fórmula principal sería:

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \times \mathrm{idf}(t, D)$$

Ecuación 6: fórmula del factor TF-IDF.

El factor de un término ***t***, en un documento ***d***, dentro de una colección ***D***, es el resultado de la función de frecuencia del termino ***t*** dentro del documento ***d***, por la frecuencia inversa del término en la colección de documentos ***D***.

Para la frecuencia del termino ***tf(t, d)*** se pueden dar varias fórmulas, la más simple es que*)* sea la frecuencia del término en el documento: ***f(t,d)***. Aunque para evitar una predisposición hacia los documentos extensos se utiliza a veces la frecuencia normalizada:

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

Ecuación 7: frecuencia normalizada de término.

En la frecuencia normalizada se divide la frecuencia del término por el valor máximo de frecuencia del término más frecuente en el documento.

Para la frecuencia inversa de documento, el término ***idf(t, D)***, se calcula tomando el logaritmo de la división del número total de documentos por el número de documentos que contienen el término.

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

Ecuación 8: frecuencia inversa de documento.

2.2.4 K-NN

Este método, llamado el de los K-vecinos más cercanos (K *nearest neighbours*) es como *SVM* un algoritmo de clasificación supervisada con aprendizaje basado en un conjunto de entrenamiento que sirve para estimar la función de densidad de las variables predictoras por cada clase C_j.

Es un método de clasificación no paramétrico que estima el valor de la función de densidad de probabilidad o directamente la probabilidad a posteriori de que un elemento *x* pertenezca a la clase C_j a partir de la información proporcionada por el conjunto de entrenamiento.

El espacio de los vectores posibles es particionado en regiones por localizaciones y etiquetas de los ejemplos de entrenamiento. Un punto es asignado a la clase C_j si ésta es la clase más frecuente entre los *k* ejemplos de entrenamiento más cercano. Para determinar los ejemplos más cercanos se suele usar la fórmula de la distancia euclidiana.

La fase de entrenamiento del algoritmo consiste en almacenar los vectores característicos y las etiquetas de las clases de los ejemplos de entrenamiento. En la fase de clasificación, se evalúa el

ejemplo del que no se conoce su clase: se calcula su distancia hacia los vectores almacenados y se selecciona los *k* ejemplos más cercanos. El vector examinado es tomado como nuevo ejemplo y es clasificado con la clase que más se repite en los vectores seleccionados.

El método supone que los vecinos más cercanos nos dan la mejor clasificación y esto se hace utilizando todos los atributos.

Uno de los problemas derivados radica en la posibilidad de que haya muchos atributos irrelevantes que terminen por decidir sobre la clasificación, así se restaría peso a los atributos relevantes. Para salvar este sesgo se opta por asignar un peso a las distancias de cada atributo, dando así mayor importancia a atributos más relevantes.

El algoritmo de entrenamiento es sencillo: para cada ejemplo se agrega el ejemplo a la estructura con la clase ya asignada.

El algoritmo de clasificación, consiste en regresar la siguiente fórmula:

$$\hat{f}(x) \leftarrow argmax_{v \in V} \sum_{i=1}^{k} \delta(v, f(x_i))$$

Ecuación 9: formula de clasificación k-nn.

Donde δ(v, $f(x_i)$) es 1 si v = $f(x_i)$ y 0 en caso contrario.

El valor k.

La elección del valor *k* depende de la naturaleza de los datos, por regla general, un valor grande de *k*, reduce el efecto de ruido en la clasificación, pero crea límites entre clases parecidas. A veces se busca un valor *k* por optimización.

Cuando *k* = 1, se llama algoritmo del vecino más cercano, se trata del caso especial en que la clase se predice para ser la más cercana al ejemplo de entrenamiento.

2.2.5 Análisis de red social SNA

Los métodos SNA analizan las relaciones humanas desde una perspectiva matemática. Mapean y miden las relaciones y los flujos de comunicación entre personas, organizaciones y otros entes conectados en una red social. En la tarea de analizar una red social de correos electrónicos en busca de los expertos o especialistas (*expertise*) se recomienda un proceso de tres pasos:

1. Recolectar todos los correos de un tema determinado de más de una palabra; en este contexto

un correo electrónico se considera relacionado con un tema si contiene al menos una palabra sobre ese tema.

2. Construir un grafo basado en el primer paso. En el grafo los nodos son contactos y las aristas dirigidas se crean por los campos ***From*** y ***To*** de los mensajes.

- Obtener puntuaciones de todos los candidatos en el grafo de expertos. Los candidatos dentro dentro de las k mejores puntuaciones son preferidos. Se recomienda usar una versión modificada del algoritmo HITS **[Kimball]**: The Data Warehouse Toolkit – Ralph Kimball, Margy Ross,
  Ed Wilet 3th edition

3. **[Kleinberg]**).

Otros métodos SNA

Existen más métodos de análisis social, algunos se basan en la frecuencia con que un contacto se relaciona con otros. Las versiones mejoradas incluyen una componente temporal para que las agrupaciones no sean rígidas y un contacto que en el pasado mantuvo una relación estrecha pero con el que hace tiempo que no se mantiene relación pueda quedar fuera de la agrupación.

2.2.6 Algoritmo Girvan-Newman

Este algoritmo es habitual en la detección de comunidades dentro de una red social, la idea general es calcular repetidamente la distancia de las aristas de una red y entonces eliminar las aristas más distantes que se satisfagan unas condiciones de aceptabilidad, **[Girvan y Newman]**. Las aristas que quedan al final del proceso son las que unen los nodos de las comunidades.

Se basa en el concepto de la intermediación de arista, (*egde betweenneess*), definida como el número de caminos más cortos entre pares de nodos que se ejecutan a lo largo de ella. Si hay más de una ruta más corta entre un par de nodos cada ruta se le asigna el mismo peso tal que el peso total todos los caminos es igual a la unidad. Si una red contiene comunidades o grupos que están sólo vagamente conectados, entonces todos los caminos más cortos entre las diferentes comunidades deben pasar por una de estas pocas aristas. Por lo tanto, los enlaces de conexión entre comunidades tendrán alta intermediación (al menos uno de ellos). Mediante la eliminación de estos enlaces los grupos están separados uno de otro, momento en el que la estructura de la comunidad subyacente en la red se revela.

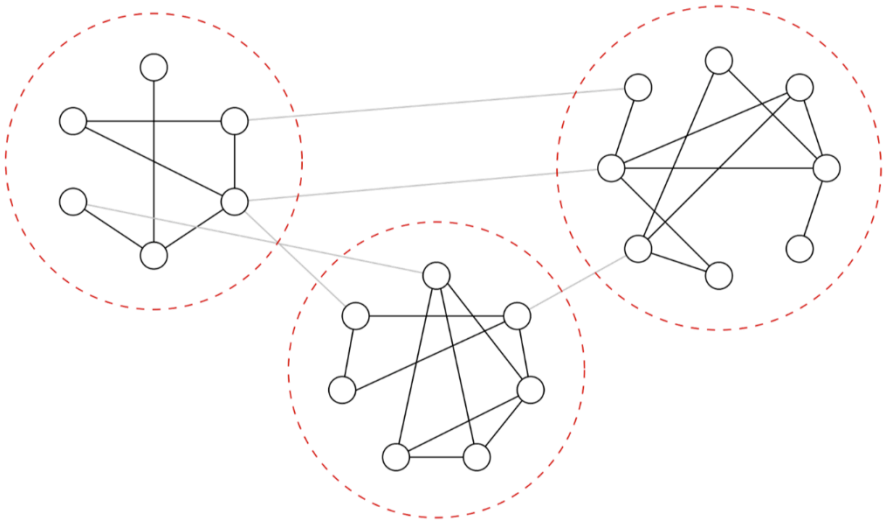


Figura 3: Ejemplo de comunidades tratadas por Girvan-Newman [Girvan y Newman 2012].

En la figura anterior los nodos dentro de cada circunferencia en rojo, son comunidades y las aristas en gris serían las que unen dichas comunidades.

Los pasos serían:

1. Primero se calcula la intermediación de todas las aristas existentes en la red.
2. Se elimina el enlace con la más alta intermediación.
3. La intermediación de todas las aristas afectadas por la eliminación se vuelve a calcular.
4. Se repiten los pasos 2 y 3 hasta que no quedan aristas.

El hecho de calcular la intermediación de las aristas afectadas por la eliminación en el paso 3, permite reducir el tiempo de computación no obstante hay que recalcular la centralidad de la intermediación para evitar que se produzcan errores graves.

2.3 Sistemas y aplicaciones para estudios relacionados

2.3.1 Snap System

Standford Network Analysis Project, **[SNAP]**, es un sistema de alto rendimiento para procesar redes de hasta billones de aristas. Es un producto de la universidad de Standford con el patrocinio de grandes empresas como Google. En la página web del grupo se puede descargar las librerías en C++ y Python bajo licencia BSD.

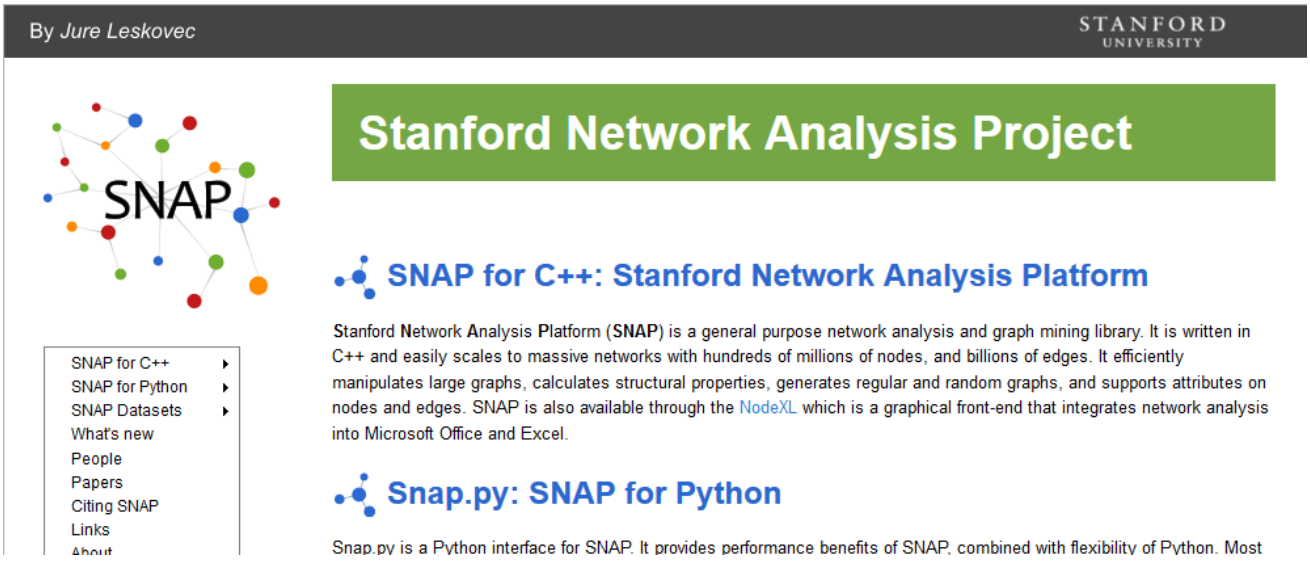


Figura 4: site web de SNAP.

El grupo presenta regularmente sus trabajos de análisis de redes sociales, en el marco de otro tipo de análisis como el de redes de grafos de grandes dimensiones, análisis de redes naturales, etc.

2.4 Herramientas de análisis de correo

Las principales herramientas de análisis de correo o bien se encuadran en administración de los sistemas de correo corporativo de una organización o bien son aplicaciones del ámbito de la minería de correo electrónico descrita anteriormente. Es evidente que el requerimiento de privacidad de los mensajes alza un obstáculo para el análisis de los correos, en especial los análisis propios de la minería de correo electrónico vistos en el apartado **[2.1Finalidades de la minería de correo electrónico]**. Algunas herramientas de gestión y MTA contienen módulos de informes que calculan las estadísticas

tratando los mensajes de un modo anónimo, sin analizar el contenido o asunto de los mismos y tomando un id alternativo opaco a las cuentas de correo.

2.4.1 Sendmail

Para el *MTA Sendmail* (Véase **[Send]**), uno de los más populares, existe la herramienta: *Sendmail Analyzer* **[SendAn]**, que permite procesar los archivos de *log* de correo electrónico y generar dinámicamente estadísticas en HTML con gráficas. Los Informes son generados en tiempo real de tal modo que permiten conocer al momento lo que sucede en los servidores de correo.

Algunas de las estadísticas que ofrece son:

Estadísticas Globales:

- Número de mensajes entrantes y salientes:
  - Mensajes regulares
  - Mensajes de *spam*
  - Mensajes con virus
- Mensajes de error de entrega y recepción (*Delivery Status Notification*)
- Mensajes distribuidos que provienen de *internet*
- Mensajes que se han enviado internamente (a la organización)
- Mensajes que se envían hacia afuera
- Mensajes que vienen y van a *internet*
- Estadísticas de Uso

Estadísticas concretas:

- El dominio de emisor y la dirección de emisor con más mensajes enviados
- Menajes recibidos:
  - El dominio receptor de más mensajes
  - Las direcciones de receptores con más mensajes recibidos
- Envío de *spam*:
  - Los dominios de envío de *spam* más activos

- las direcciones de envío más activas
- Las direcciones destinatarias más activas

- Los números máximos de destinatarios para un mensaje

Los tipos de informe que ofrecen son HTML con gráficas prácticas aunque ofrecen una parametrización escasa. El eje temporal que manejan suele ser el mes.

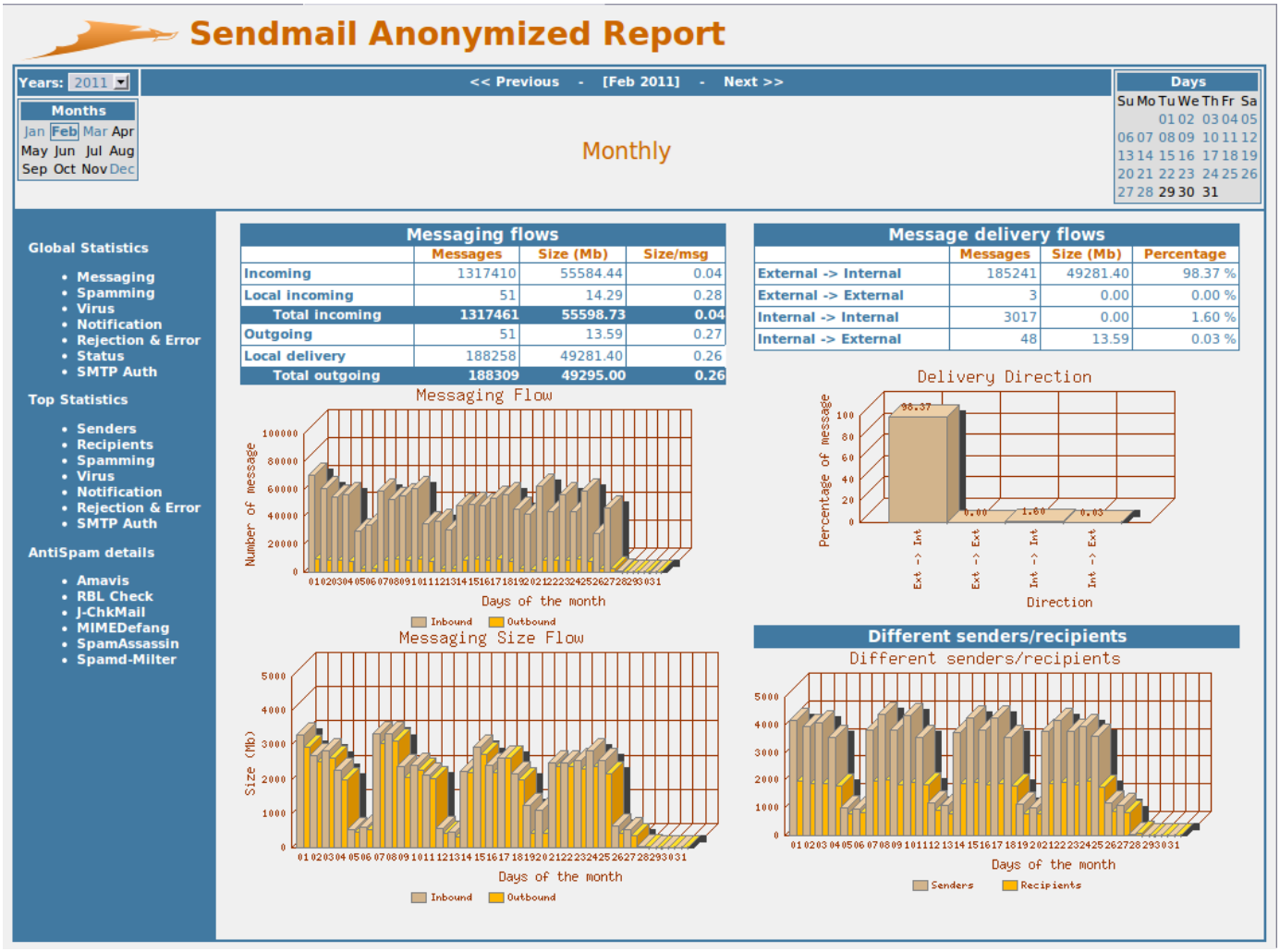


Figura 5: Informe tipo de la herramienta Sendmail Analyzer.

2.4.2 Gmail Meter

Es una herramienta de reporte estadísticos para cuentas de Gmail y apps Google (véase **[Gmail Meter]**). Permite generar informes cada semana con estadísticas detalladas y reporte analíticos a nivel personal de como un usuario utiliza Gmail.

Entre los informes que ofrece se encuentran reportes de volumen horario y semanal, los emisores y receptores en el *top*, longitudes de hilo o tiempo medio de respuesta.

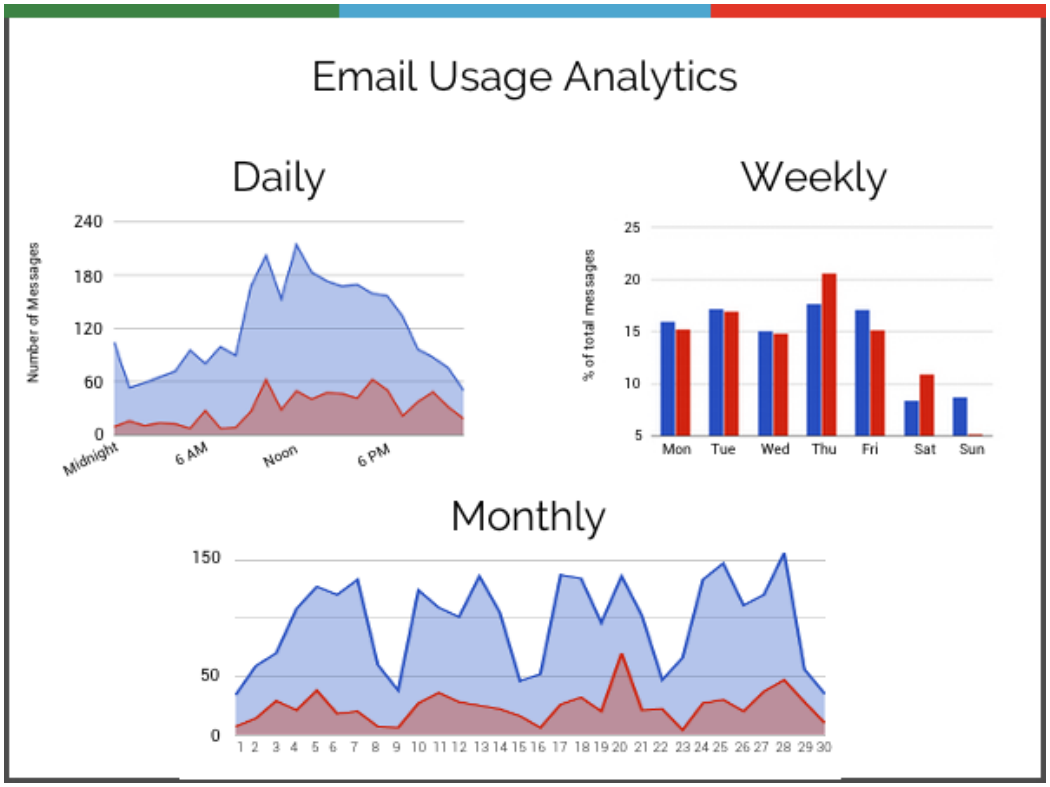


Figura 6: Estadísticas más populares de Gmail Meter.

La aplicación también permite tener reportes de tiempos de respuesta, tendencias diarias y semanales y descubrir patrones de uso orientado a la eficiencia en el uso del correo electrónico:

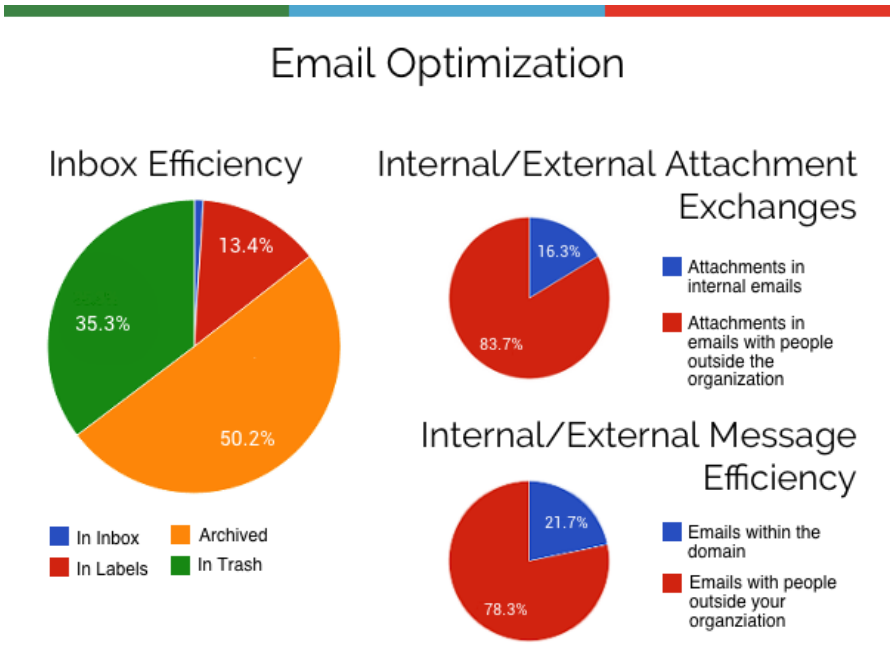


Figura 7: Estadísticas de optimización de Gmail Meter.

Su principal limitación es que se utiliza con un alcance personal lo que no aporta mucho valor como un origen de datos para explotar. No obstante la filosofía y la finalidad de algunos de sus reportes si son interesantes en el alcance de este proyecto fin de carrera.

2.5 Conjuntos de datos públicos y privados

El número de conjuntos de datos públicos de correo electrónico es muy escaso, el único realista es Enron. Algunos otros de *spam y phising* están también disponibles aunque requieren el permiso de los investigadores y empresas que los han recolectado.

Enron Email Dataset

La fuente más recurrente está en: https://www.cs.cmu.edu/~./enron/, son los correos electrónicos hechos públicos de Enron, recopilados para la iniciativa CALO: un proyecto de asistente cognitivo para organizar y aprender. Los mensajes se hicieron públicos en internet por la Comisión Regulatoria de la Energía de Estados Unidos durante su investigación del caso de ENRON.

El conjunto de datos original contiene en torno a medio millón de mensajes de unos 150 usuarios, aunque algunos investigadores contribuyeron a limpiar el conjunto de datos reduciendo su número.

No incluye adjuntos y algunos mensajes han sido eliminados ante la solicitud expresa de varios empleados afectados.

Una versión más actualizada del conjunto de datos se puede consultar en http://www.edrm.net/resources/data-sets/edrm-enron-email-data-set, donde se incluyen los ficheros adjuntos.

Phising files

Es conocido por ser un conjunto de datos "malo" es decir contiene una colección de correos electrónicos de *phising* utilizado en diversos estudios. Ha sido recopilado por el doctor José Nazario (http://monkey.org/~jose/wiki/doku.php?id=phishingcorpus). En la actualidad hay que solicitar permiso para utilizarlo.

Gmailmeter

Más que un conjunto de datos, es una herramienta para generar *datasets* de prueba en base al correo personal o a cuentas de correo usadas con el permiso de sus propietarios. *GmailMeter* además de ofrecer una herramienta de análisis permite extraer los mensajes de correo de las cuentas de usuario.

3 ESTÁNDARES Y PROTOCOLOS DEL CORREO ELECTRÓNICO

Se presenta un resumen de los principales estándares y protocolos que intervienen en la transmisión y almacenamiento del correo electrónico y como se relacionan.

En este capítulo se presenta por un lado los protocolos y la arquitectura habitual de servidores y terminales que intervienen en el envío y recepción de los mensajes y por otro la gramática de los elementos más importantes a nivel léxico y sintáctico que participan en la construcción de los mensajes. En este segundo apartado, el estudio de los detalles gramaticales de las direcciones de correo, grupos de direcciones, mensajes y cabeceras constituyen los fundamentos a la hora de construir consultas y filtros de bajo nivel.

3.1 Visión general de los Protocolos

Los protocolos que se utilizan en el campo del correo electrónico, son materia curricular de la ingeniería informática, ciencias de computación y telecomunicaciones. Esta sección no pretende ser más que un breve repaso de los aspectos que más pueden influir para el estudio de un conjunto de correos electrónicos.

Empezaremos viendo la arquitectura básica de la comunicación mediante correo electrónico, **[Stevens]**:

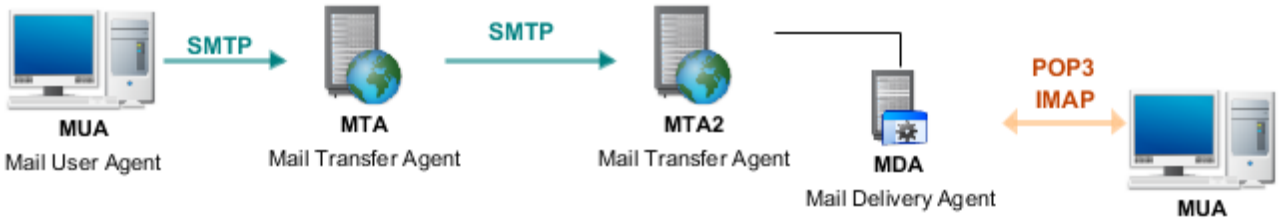


Figura 8: Visión general de protocolos del correo electrónico.

En el escenario más habitual de la comunicación por correo electrónico, un usuario utiliza su cliente de correo o **MUA** (*Mail User Agent*) para enviar un mensaje. En otros tiempos este agente trabajaba junto a un agente de entrega de correo o **MDA** (*Mail Delivery Agent*) aunque en la actualidad podemos considerar esas funcionalidades integradas en el propio cliente de correo.

El mensaje se envía a un agente de transferencia de correo o **MTA** (*Mail Transfer Agent*) que será el encargado de enviar el mensaje al siguiente nodo de la red. Generalmente aquel será otro **MTA** pero en este caso perteneciente a la corporación del destinatario. Nos referimos a este agente simplemente como servidor de correo.

En la comunicación entre cliente y servidor (MUA-MTA) se emplean los protocolos **POP3** e **IMAP**. En la comunicación entre los servidores de correo se utiliza el protocolo **SMTP** (*Simple Mail Transfer Protocol*) aunque las RFC admiten que el protocolo es de aplicación para la red pública de envío de correo, pudiendo usarse un protocolo propietario distinto en la red de correo interna de una organización.

En un escenario alternativo el usuario se conectaría por un navegador web a un sitio de *webmail*. En este contexto, el sitio web actuaría como cliente de correo (**MUA**) permitiendo al usuario la edición y la visualización de correos y también como **MTA** transfiriendo el mensaje vía **SMTP**.

3.1.1 MUA: Clientes de correo

Son programas de software que se emplean para transferir el correo desde el terminal del usuario a un servidor de correo electrónico. Funcionan en ambos sentidos: permiten al usuario editar y enviar un mensaje, pero también revisar su bandeja de correo entrante.

Se suele operar persistiendo los mensajes recibidos en los buzones de correo en un formato concreto de fichero físico. Los formatos más usados para almacenar los mensajes son ***mbox***, cuyo contenido se importa en la aplicación adscrita a este proyecto y ***maildir***.

El formato de almacenamiento del buzón es muy relevante de cara a importar un conjunto de datos que siga esa misma estructura de ficheros. Otros aspectos propios de los MUA como las cabeceras extendidas y cabeceras opcionales pueden afectar a los análisis que se hagan de ellos.

En la fecha de redacción de esta memoria Wikipedia ya lista 38 clientes de correo con interfaz gráfica y 16 clientes web. Casi medio centenar de clientes de correo sin contar clientes de uso histórico ni los que están basado en texto. Con un número tan grande no merece la pena pormenorizar los casos concretos, a menos que sea un requisito explícito.

Formato de fichero de buzón *mbox*

Es un término genérico para una familia de formatos de documento que se usa para almacenar conjuntos de correos electrónicos. Todos los mensajes de un buzón de correo (*mailbox*) quedan concatenados en un único fichero. El principio de cada mensaje está marcado por una línea que empieza por los cinco caracteres ***«From »*** continua por el texto del mensaje y termina en una línea en blanco para marcar el final.

Durante un tiempo el formato ***mbox*** fue popular debido a que se podía usar muy fácilmente herramientas para el procesado de documentos de texto y modificar dichos documentos. Aunque presenta como principal desventaja que procesar el fichero puede requerir algo de tiempo durante el cual ha de quedar bloqueado para evitar que varias escrituras concurrentes lo puedan dejen en un estado inconsistente.

Al contrario de los protocolos de Internet usados para el intercambio de correo, el formato usado para almacenamiento de correo se dejó completamente en manos del desarrollador del cliente. El formato ***mbox*** nunca ha sido formalmente definido a través de una RFC, por lo que han aparecido programas de conversión para transferir el correo entre los distintos clientes. El hecho de que desde los principales clientes de correo como *Gmail*, *Outlook* y *Thunderbird* se pueda exportar a este formato, lo han convertido junto a EML en un estándar de facto para la exportación de buzones de correo.

Hay variantes del formato ***mbox***, siendo las más representativas: MBOXO, MBOXRD, MBOXCL y

MBOXCL2 que cambian en los caracteres de la línea separadora ***«From »*** y en el uso de etiquetas ***Content-length***. Es importante remarcar que estas versiones son incompatibles entre sí.

Maildir

Es un formato de *spool* de correo electrónico que no bloquea los ficheros para mantener la integridad del mensaje; los mensajes se almacenan en ficheros distintos con nombres únicos. Trabaja sobre un directorio (usualmente llamado *maildir*) con tres subdirectorios: *tmp*, *new*, y *cur*. Todos los subdirectorios deben residir en el mismo sistema de archivos. Sigue una estrategia de enlazar primero y desenlazar al mover los ficheros desde la carpeta de *temp* a *new* y luego a *cur*.

EML

No es un formato de almacenamiento de buzón como los que presentamos en esta sección, pero es el formato típico para exportar correos de manera individual. No tiene una definición estándar propia aunque se lo considera una extensión del formato *IMF* (*Internet Mail Format*) con el que deben trabajar las **MTA** al transferir mensajes entre máquinas, (véase **[RFC 5322]**).

PST

Sus siglas vienen de *Personal Folders File*, es un formato propietario abierto (**[Digital Formats PST]**) que se usa para almacenar copias locales de mensajes, eventos de calendario y otros elementos dentro del software de *Microsoft Office Outlook*.

Existen dos versiones: *PST-ANSI* y *PST-Unicode* siendo la mayor diferencia que la última permite tamaños de fichero de hasta 50 GB en las versiones 2010 y 2013 de *Outlook* y que emplea el juego de caracteres *Unicode*.

Estos ficheros se estructuran en dos árboles B que indexan una arquitectura de tres capas:

- Capa NDB (*Node Data Base*) que almacena los bloques físicos de almacenamiento, consiste en cabecera, información de almacenamiento, bloques, nodos y dos árboles B:
    - NBT *(Node B Tree*)
    - BBT (*Block B Tree*) que implementa la asignación de almacenamiento dentro del fichero PST
- Capa LTP (*List, Tables and Properties*) que implementa conceptos de alto nivel dentro de la estructura lógica como el Contexto de Propiedades, el Contexto de Tabla y en general colecciones de propiedades.
- Capa de Mensajes (a veces nombrada como la capa **PST**) que implementa los objetos de directorio, objetos de mensaje, etc… como una estructura de listas, tablas y propiedades.

Afortunadamente las librerías de interoperación de *Office* permiten acceder a la mayoría del contenido de los ficheros PST sin tener que tratar a bajo nivel con las capas o los árboles B. Para implementar la importación desde *Outlook* éste ha sido el enfoque empleado.

3.1.2 MDA: Distribuidores de correo

Es el programa que filtra el *spam*, ordena el correo y lo distribuye a los buzones propios del usuario según las reglas que haya definido. En la actualidad es una parte muy integrada en el MTA, recibiendo el nombre de LDA (*Local Delivery Agent*). El MTA se limita a invocar al LDA ante la llegada de un fichero de correo nuevo.

En sistemas *Unix* los más utilizados son *procmail* y *maildrop*.

3.1.3 MTA: Servidores de Correo

Los agentes de transferencia de correo MTA, o sencillamente servidores de correo, se usan para transferir mensajes entre máquinas. Los clientes de correo MUA envían sus mensajes a su MTA, que habitualmente será el configurado como servidor de correo saliente, el MTA se encarga de redirigir el mensaje al siguiente MTA que permita hacer progresar al mensaje hasta el buzón del destinatario.

Los usuarios podrían acceder a sus correos directamente sin tener que hacer el paso Usuario-MUA-MTA, aunque requiere conocimientos avanzados para poder comunicarse por comandos con el MTA y por eso solo unos pocos usuarios expertos accederán así.

Algunos servidores de correo pueden incluir cabeceras extendidas propias que sean de interés a la hora de determinar ciertas características del corpus de correos electrónicos, por ejemplo algunos campos de direcciones extendidas de *Lotus Notes* en el caso de Enron.

Un listado de MTAs sería:

- *Sendmail*
  - https://www.**sendmail**.com, constituye el programa de MTA *más popular*
- *Gmail*
  - https://mail.google.com/, con una extensa gama de servicios web
- *Postfix*
  - www.**postfix**.org/
- *Exim*:
  - www.**exim**.org/, *desarrollado por la universidad de Cambridge*

- *Mdaemon*
  - www.mdaemon.es
- *Mercury Mail System*
  - www.pmail.com, para sistemas *Novell*
- *Lotus Notes (IBM Notes)*
  - www.ibm.com/software/products/es/ibmnote
- *Microsoft Exchange Server*
  - https://www.microsoft.com

3.1.4 SMTP: La comunicación entre Servidores de Correo

Sus siglas vienen del inglés: *Simple Mail Transfer Protocol*, dicta el modo en que los servidores de correo MTA envían los mensajes. Se define en la **[RFC 821]** y consta solo de una docena de comandos sobre TCP/IP. Es relevante destacar que durante el enrutamiento que hace un MTA se toma la dirección dictada por el servidor de origen a través de SMTP y no por la dirección que figura en el mensaje.

Posteriores revisiones del protocolo constituyen el llamado ESMTP (Extended SMTP: **[RFC 1425]**, **[RFC 1651]**). En el año 2005 y después en 2008 (**[RFC 5321]**) se revisó el protocolo agregándole funciones adicionales en el denominado XSMTP que supone la versión más habitual en la actualidad. Otra versión del protocolo llamada SMTPS utiliza SSL para comunicaciones más seguras.

Las mejoras al protocolo han ido en la dirección de salvar las restricciones sobre la codificación NAVT ASCII impuesta en la**[RFC 821]**, hasta llegar a la flexibilidad de la especificación MIME enriqueciendo características de usabilidad y semántica.

Aunque algunos servidores y clientes utilizan SMTP para consultar el correo no es lo habitual, para ello son mucho más populares los protocolos POP3 e IMAP. Este protocolo es el estándar para la comunicación entre sistemas heterogéneos, para la comunicación interna de sistemas propietarios como *MS Exchange*, *Lotus Notes*, *Outlook.com* y *Gmail, se* usan protocolos propios.

Por último mencionaremos que el IETF trabaja en la **[RFC 6531]** para sobrepasar el límite del juego de caracteres NVT ASCII en los requisitos de la dirección de correo, es la extensión SMTPUTF8 que soporta caracteres *multibyte* en las direcciones de correo. Direcciones como Pelé@live.com (latín con acentos diacríticos), δοκιμή@παράδειγμα.δοκιμή, y 测试@测试.测试 serán direcciones válidas. Aunque a día de hoy no está muy extendido supone una opción de futuro muy plausible.

3.1.5 MIME

Es un grupo de extensiones a los protocolos de correo electrónico, (*Multipurpose Internet Mail Extensions*) recogido en varias RFC: **[RFC 2045]**, **[RFC 2046]**, **[RFC 2049**]. Se permite el envío de objetos adicionales al texto plano como vídeo y audio; y se implementan mecanismos para agregar partes de distinta naturaleza a un mismo mensaje. Consigue que se superen las restricciones de longitud de línea y de la codificación en 7-*bits* de los textos para poder internacionalizar los mensajes aunque extensiones previas a SMTP ya habían hecho ciertos avances en ese sentido.

De un modo resumido definen un mecanismo para poder separar la codificación del contenido del correo de la codificación de la transmisión del mismo facilitando además codificaciones heterogéneas de cada parte del contenido. Para objetivos de análisis de correos electrónicos es muy importante saber reconocer la semántica y poder interpretar correctamente las partes codificadas con MIME.

Las cabeceras estándar que se incluyen son:

- ***MIME-Version***: su presencia indica que el mensaje utiliza el formato MIME, siempre se usa la versión 1.0.
- ***Content-Type***: Indica el tipo de medio que representa el contenido del mensaje, se expresa como *type/subtype*, por ejemplo *text/plain*.
- ***Content-Transfer-Encoding***: Indica el método de codificación para la transferencia del mensaje:
    - *7bit*: Es el valor implícito y utilizado en la especificación original de SMTP
    - *Quoted printable*: Se utiliza para codificar secuencias para que se cumpla la regla de codificación de *7 bit*.
    - *Base64*: usado para codificar secuencias arbitrarias de octetos para que se cumplan los requisitos de *7 bits*.
    - *8 Bit*: si se utiliza la extensiones SMTP (8BITMIME)
    - *Binary*: usado para servidores SMTP que acepten la extensión SMTP BINARYMIME
- ***Content-ID*:** permite identificar cada parte de un mensaje mime para poder referenciar partes en otros lugares del mensaje.

A continuación presentamos una tabla con los tipos de contenido de MIME.

Type	*Subtype*	Notas
text	*plain*	Texto sin formato.
	richtext	Texto con formato simple como negrita, cursiva, subrayado etc.
	enriched	Un versión más refinada de *richtext*
multipart	*mixed*	Múltiples partes para ser procesadas secuencialmente.
	parallel	Múltiples partes para ser procesadas en paralelo.
	digest	Una manera de enviar múltiples mensajes de texto. Implícitamente cada parte es *message/rfc822*.
	alternative	Indica que cada parte es una versión alternativa del mismo contenido (por ejemplo una imagen y la descripción de la misma).
	related	Se usa para indicar que las partes son agregados de un todo. Se suele utilizar para enviar páginas web completas con imágenes en un único mensaje. La parte raíz contendría el documento HTML que se referiría a las imágenes almacenadas en las partes siguientes.
	report	Contiene datos formateados para que un servidor de correo lo interprete.
	signed	Se utiliza para adjuntar una firma digital al mensaje.
	encrypted	El mensaje tiene dos partes, la primera contiene información de control que es necesaria para descifrar la segunda parte de tipo *application/octet-stream*.
message	*rfc822*	El contenido es otro mensaje **[RFC 822].**
	partial	El contenido es un fragmento de correo electrónico.
	external-body	El contenido es un puntero al mensaje actual.
application	[…]	Se han definido muchos tipos de aplicaciones dentro de MIME, el tipo más neutro sería *Octet-stream* que indica datos binarios arbitrarios.
image	[…]	Se han definido distintos formatos de imagen: *jpeg, gif, png, bmp*…
audio	[…]	Audio codificado de diversos tipos.
video	[…]	Diversos formatos de video (*mpeg*: *ISO 11172*, *mp4*, etc…).

Tabla 1: Tipos de contenidos MIME.

Un aspecto de especial interés de MIME es la división en secciones, mediante la cual se pueden introducir en un mismo mensaje de correo distintos tipos de contenido. Con la cabecera ***Content-Type***=”*multipart*” se puede definir un *token* de límite mediante ***Boundary***=”<*tokenlímite*”> que diferenciaría varias partes dentro del cuerpo del mensaje. Cada una de esas partes puede tener a su vez una cabecera **Content-Type** que aplica sobre la sección permitiendo así tener distintas secciones anidadas cada una de un tipo distinto.

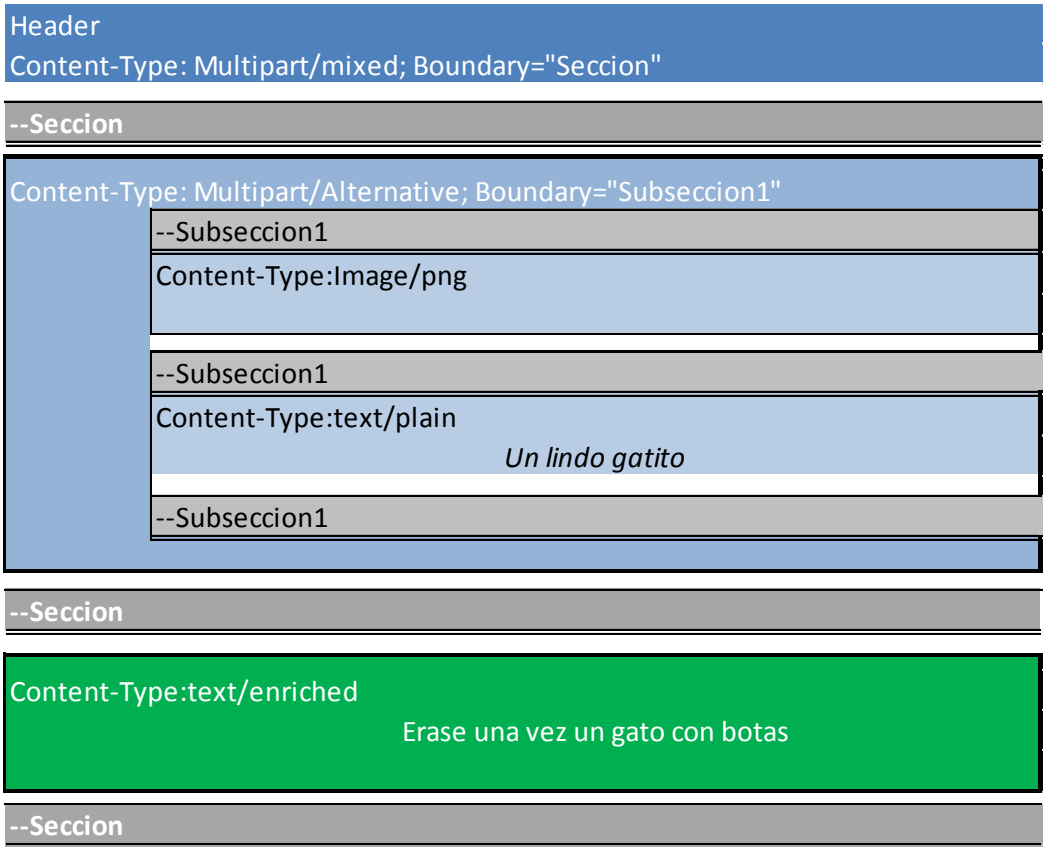


Figura 9: Esquema de un mensaje MIME multi-parte.

El subtipo “*Alternative*” indica que las dos partes son alternativas si el cliente de correo solo permite mostrar texto se presentará la sección de texto plano “Un lindo gatito” si muestra imágenes se presentaría la primera subsección.

3.1.6 Otros Protocolos

IMAP y POP3

Estos protocolos definen como interactúa el terminal del usuario final frente a un servidor de correo, para servir el mensaje hasta el terminal del usuario. POP3 está recogido en **[RFC 1939**].

HTML y CSS

Son los estándares que definen la estructura y presentación de las páginas web que se sirven a los usuarios. Son relevantes en el análisis y estudio del correo electrónico, porque están muy extendidos como formato de mensaje en forma de plantillas y pueden definir una semántica propia dentro de los correos de un grupo de usuarios que usan la misma plantilla de correo HTML.

Se puede identificar los mensajes con etiquetas HTML porque deberían tener el tipo MIME: *text/html*. Si se quiere hacer análisis con el contenido del mensaje, por ejemplo desde la perspectiva del lenguaje natural, habría que extraer del cuerpo el contenido que el usuario lee, de la estructura de etiquetas HTML.

3.2 Estructura Léxica y Sintaxis de los mensajes de correo

3.2.1 Listado de RFC

Vamos a repasar las RFC más relevantes para la interpretación léxica, sintáctica y semántica de los mensajes de correo electrónico.

- **[RFC 822]** ***Standard for Format of the ARPA internet text messages*** *(Agosto de 1982):* Constituye la base para el formato de los ficheros de correo electrónico, es la primera RFC que trata del formato de los mensajes de texto en el contexto del correo electrónico.
- **[RFC 2822]** ***Internet Message Format*** (Abril 2001): actualiza los estándares de la anterior RFC declarando obsoletas algunas formas de dirección de correo electrónico.
- **[RFC 5322]** ***Internet Message Format*** (Octubre 2008) deja obsoleta a la anterior 2822 y actualiza parte del contenido relativo a cabeceras MIME.

Al emplear métodos de minería de correo electrónico y en especial con corpus antiguos, hay que tener en cuenta las normas definidas por las RFC, el estándar de facto asumido por los principales sistemas de correo en la época. En el corpus de Enron por ejemplo la **[RFC 5322]** aún no estaba vigente. Este punto supone una dificultad añadida en algunos estudios, por ejemplo las reglas para clasificar como *spam* entrenadas con corpus antiguos pueden perder cualidades al ser utilizadas por

sistemas actuales.

En cada definición, el punto principal consiste en la definición de las direcciones de correo empleada en muchos tipos de estudios: clasificación de correos, análisis de redes sociales, etc.

3.2.2 Estructura básica:

Los mensajes de correo electrónico son ficheros de texto plano con codificación US-ASCII (inicialmente, luego se ve modificada por extensiones de servicio SMTP y por MIME) en los que se distingue dos partes claras de un lado la cabecera del mensaje y del otro el cuerpo que es opcional: pueden existir mensajes con la cabecera y el cuerpo vacío.

Las cabeceras son líneas de texto delimitadas por el grupo retorno de carro y nueva línea, tienen una longitud máxima de 998 caracteres, aunque la longitud recomendada para el ancho de pantalla de su época es 78 (80 con el retorno de carro y la nueva línea). Pueden presentarse cabeceras más largas con un proceso especial que se denomina *folding* para hacer que la cabecera se extienda entre varias líneas y *unfolding* para recuperar el contenido de una cabecera a la que se ha hecho *folding*. Algunas extensiones de la RFC original permiten emplear líneas de texto con mayor longitud.

Los caracteres especiales, dentro del ASCII con el que trabaja el protocolo son:

- Paréntesis: la apertura y cierre de paréntesis '**(**' '**)**' son utilizados en los comentarios.
- "*Angle brackets*", los signos menor y mayor '**<**' y '**>**', se utilizan para delimitar la parte de una dirección del texto del buzón en una de las formas de dirección de correo electrónico permitidas: *John Doe* <*user.name@server.com*>.
- Arroba, '**@**' que diferencia la parte de usuario de la parte de servidor en una dirección de correo. También se usaba en la **[RFC 822]** en el *token route* para indicar una ruta de dominio y en la **[RFC 2822]** en el *token* ***Msg-ID*** para formular los ID de mensaje.
- Nueva línea y retorno de carro (**CRLF**) son los espacios en blanco que introducen una nueva línea. Debido al modo en que el protocolo define el fin de campos en las cabeceras, el *folding*, etc…, el retorno de carro se vuelve importante para aceptar e interpretar un mensaje.
- Cadenas Entrecomilladas, **"***cadena***"**, para escapar una serie de caracteres especiales, basta con poner la cadena entre comillas, así por ejemplo en: "Philip K. Dick" <p.k.dick@mailserver.com> podemos usar el punto y el espacio en el nombre del buzón.

- Carácter de Escape: Se usa la *contrabarra* '\' para escapar un carácter que de otro modo no sería aceptado por el protocolo. Se permite de modo particular cuando se utilizan otro par de caracteres delimitadores así por ejemplo dentro de una cadena entrecomillada "cadena" se permite introducir el carácter comilla doble si va escapado por la *contra-barra*, la propia *contra-barra* se puede escribir como doble contra-barra "\\".
- Signos de Puntuación, el punto "." se utiliza dentro de direcciones de correo electrónico tanto en la parte de nombre como en la parte de servidor, la coma "," y el punto y coma ";" se utilizan para dividir diversos elementos: la coma como separador general en las listas, y punto y coma para los grupos de direcciones en el *token* ***address***, tanto en la **[RFC 822]**, como en la **[RFC 2822]**. Los dos puntos ":" separan el nombre del cuerpo de los campos en las cabeceras y también para separar el ***display-name*** de la lista de direcciones de un grupo.

3.2.3 *Folding* y *Unfolding*

La definición de los *tokens* y del proceso que destacan las RFC puede resultar confusa la primera vez que se aborda. Las cabeceras se separan entre sí por los caracteres de retorno de carro y nueva línea (***CRLF***) para permitir una lectura más clara de las cabeceras. Para "escapar" de algún modo esos retornos de carro y que al interpretar un mensaje no se tengan errores en las cabeceras se habilita el mecanismo del *Folding* y *Unfolding*.

Se introduce un *token* especial: **FWS** que se compone en su definición más sencilla de un *CRLF* seguido de un espacio en blanco *WSP* " ", opcionalmente se puede incluir cualquier número de espacios por delante del *CRLF* y por detrás. Serían válidas las siguientes secuencias:

CRLF WSP

CRLF WSP WSP

WSP CRLF WSP WSP WSP

Al hacer *folding* de una cabecera podríamos pasar de:

```
[…]
To:john.doe@mailserver.com<CRLF>
Subject:<WSP>Abracadabra<CRLF>
[…]
```

a:

```
[…]
To:john.doe@mailserver.com<CRLF>
Subject:<WSP>Abracada<WSP><CRLF>
<WSP>bra<CRLF>
[…]
```

Lo relevante es que al interpretar la cabecera ***Subject***, se haría el paso contrario de *unfolding* y el valor interpretado sería *Subject = "Abracadabra"*. Este método, se diseñó para poder tener cabeceras de más de 998 caracteres, y para los primeros terminales en los que los correos se podían

leer directamente del fichero de buzón. En estos casos es de gran ayuda poder dar un formato más legible para el destinatario a la lista de direcciones, el asunto, las trazas de envío etc.

3.2.4 Especificación de Direcciones

Vamos a ver en detalle la especificación de las direcciones. En las especificaciones: **[RFC 822]**, **[RFC 2822]** y **[RFC 5322]** se tiene que una dirección puede ser un buzón (*mailbox*) o un grupo de direcciones.

El buzón puede ser a su vez una construcción de tipo nombre y dirección o una dirección simple; para el grupo de direcciones se utiliza un nombre (*display-name*) seguido de dos puntos y una lista de buzones separados por comas.

De manera esquemática tenemos:

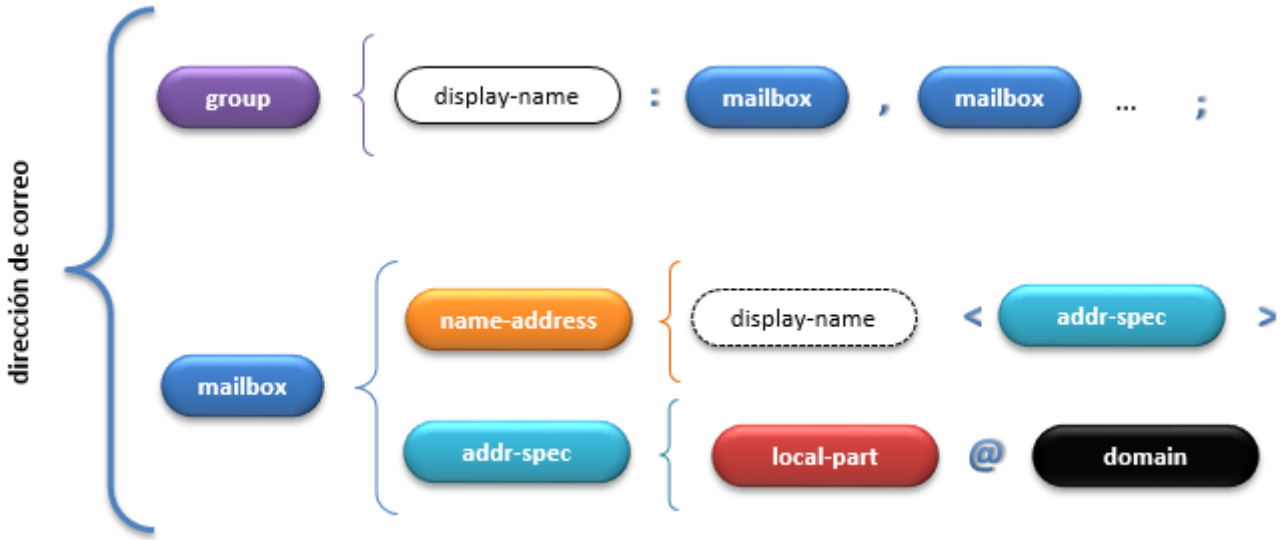


Figura 10: Sintaxis de las direcciones de correo (1 de 2).

Una dirección puede ser de grupo o un buzón, en el caso de ser una dirección de grupo se tratará de un nombre descriptivo (*display-name*) seguido de una lista separada por comas de buzones (*mailbox*) de correo.

Un buzón de correo admite dos especificaciones: por un lado una más detallada (*name-address*) que permite indicar un nombre de destinatario opcional o un texto descriptivo seguido de la dirección simple (*addr-spec*) delimitada por "*angle-brackets*" esto es los signos de menor y mayor "<" y ">".

Las direcciones simples son las más habituales y se componen de una parte local y de un dominio, se admiten puntos y caracteres ASCII comunes pero no espacios.

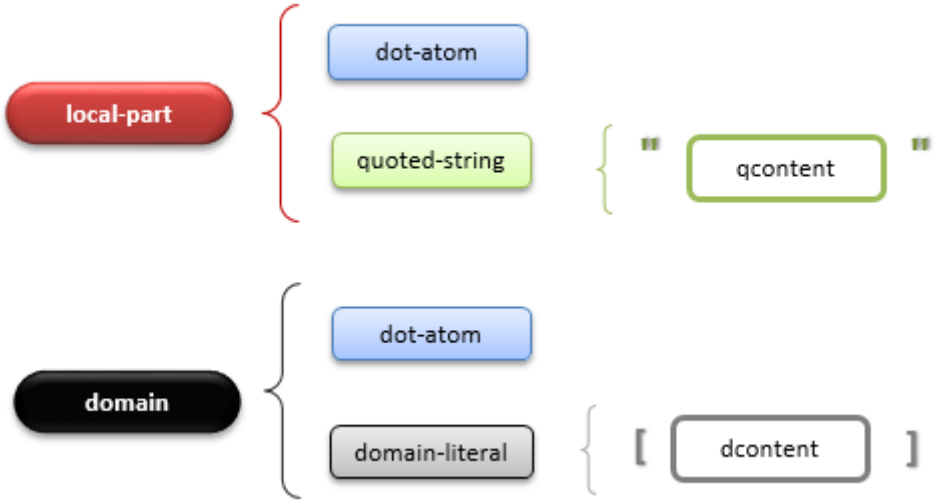


Figura 11: Sintaxis de las direcciones de correo (2 de 2).

Como método de extensión para poder utilizar más tipos de caracteres se define una forma entrecomillada para la parte local (*qcontent*) donde se puede incluir cualquier carácter incluso las propias comillas dobles y el símbolo de escape "\" siempre y cuando se escapen correctamente, así por ejemplo en: *"john.doe\"jd\""@mailman.com* .

Para los dominios se especifica un mecanismo similar que consiste en separar mediante corchetes el dominio.

3.3 Campos de Cabecera

3.3.1 Campos del Origen

Orig-date

Especifica la fecha y hora a la que el autor indicó que el mensaje estaba completo y listo para entrar en el sistema de entrega de correo: cuando pulsó el botón de "enviar". En cualquier caso no se considera la hora a la que el mensaje se transporta.

From

Especifica el autor o autores del mensaje, esto es los buzones de correo de las personas o sistemas responsables de la escritura del mensaje.

Sender

Especifica el buzón del agente responsable de la actual transmisión del mensaje. El ejemplo más típico es el de un asistente que va a enviar un mensaje de un directivo; el buzón del asistente debería aparecer en el campo ***Sender*** y el buzón del directivo, autor intelectual del mensaje, aparecería en el campo ***From***. Si ambos son la misma persona, y por ende los mismos buzones, el campo ***Sender*** no debería ser utilizado, con la información del campo ***From*** bastaría.

Reply-To

Señala una dirección de respuesta a la que enviar una contestación al mensaje, es por decirlo así la dirección sugerida para la respuesta del mensaje por parte del autor. En ausencia de esta cabecera las respuestas deberían enviarse a los buzones indicados en la cabecera ***From***.

3.3.2 Campos de Destinatarios

Especifican las direcciones de quienes recibirán los mensajes, cada uno de los campos de destinatario puede tener una o más direcciones, la diferencia de cada campo depende en como son utilizados.

To

Contiene la dirección o direcciones que suponen el destino principal del mensaje.

Cc

Significa copia carbón (*Carbon Copy*), rememorando el modo de hacer una copia al redactar una carta en una máquina de escribir con papel carbón. Dado que el campo ***To*** puede contener varias direcciones el sentido del ***Cc*** es indicar direcciones de personas que no son el destinatario principal del correo pero a las que se mantiene informadas sobre el asunto del mismo.

Bcc

Significa copia al carbón ciega (*Blind Carbon Copy*). Contiene las direcciones de los destinatarios del mensaje cuyas direcciones no son reveladas a otros destinatarios del mensaje.

Hay tres modos en que el campo ***Bcc*** puede usarse:

- Cuando un mensaje con un campo ***Bcc*** se prepara para ser enviado, el campo con las direcciones en copia oculta se elimina de todos los destinatarios, incluido aquellos a los que se envía la copia del mensaje.
- En el segundo caso: los destinatarios especificados en ***To*** y ***Cc*** reciben una copia del mensaje sin el contenido de la línea ***Bcc***, mientras que los receptores en copia oculta reciben una copia donde su dirección figura como única dirección en el ***Bcc*** y los campos ***To*** y ***Cc*** están vacíos.
- Por último el campo ***Bcc*** puede estar presente pero vacío lo que indica que el mensaje ha sido enviado con copias ciegas pero no se indica a que direcciones.

3.3.3 Campos de Identificación

Aunque opcionales en las distintas especificaciones RFC, los mensajes deberían tener campos de identificación: ***Message-ID*** y para las respuestas deberían tener ***In-Reply-To*** y ***References***.

Message-ID

Contiene un único identificador de mensaje que se refiere a una versión particular de un mensaje concreto. La unicidad del identificador del mensaje se garantiza por el terminal que lo genera. Es un identificador orientado a ser leído a nivel de máquina y no a nivel de humano.

La sintaxis de este campo es una versión limitada de la construcción *addr-spec* donde solo se permite la rama *dot-atom-text* delimitado por “<” y “>”.

In-Reply-To

Mantiene el identificador original del mensaje, se debe usar para identificar el mensaje (o mensajes) de los que el nuevo mensaje supone una contestación. Cuando hay más de un mensaje para el que es contestación este campo contendrá todos los identificadores ***Message-ID***, si no hay ningún ***Message-ID*** en ninguno de los mensajes padre, entonces el nuevo mensaje no tendrá campo ***In-Reply-To***.

References

Similar al anterior ***In-Reply-To***, se debe utilizar para identificar un hilo (*thread*) de conversación. Debe contener los campos ***References*** de los mensajes padre seguidos por los ***Message-ID*** de los padres. Si el mensaje padre no contiene un campo ***References***, pero tiene un ***In-Reply-To*** conteniendo un identificador de mensaje simple entonces el campo ***References*** tendrá el contenido del ***In-Reply-To***

del padre seguido por el contenido del ***Message-ID*** del padre. Si el padre no tiene ***References***, ***In-Reply-To*** o ***Message-ID***, entonces el nuevo mensaje no tendrá campo ***References***.

3.3.4 Campos Informativos

Estos campos son todos opcionales y contienen información destinada al usuario más que a los sistemas de correo.

Subject

Indica el asunto del mensaje, una descripción resumida del tema tratado en el mensaje. Cuando se usa en una contestación debería comenzar por la cadena "*Re:*" una abreviación del latín "*in re*". Múltiples contestaciones no deberían acumular el prefijo "Re:" en el asunto. Aunque no se indica en la **[RFC 5322]** también es común el empleo del prefijo "*Fwd:*" delante del asunto para indicar un reenvío.

Comments

Contiene comentarios adicionales al texto del cuerpo del mensaje.

Keywords

Contiene una lista de valores separados por comas de palabras y frases que pueden ser útiles para el destinatario a la hora de buscar en sus mensajes.

3.3.5 Mensajes de Traza

Los mensajes de traza son un grupo de sub-mensajes de cabecera que consisten en un campo ***Return-Path*** y uno o más campos ***Received***. La especificación completa de los campos de traza se discuten en la **[RFC 5321]**, en general el valor de los campos de traza es meramente informacional.

Return-Path:

Es un campo opcional que contiene una dirección de correo con la sintaxis *addr-spec* delimitado por lo signos de menor y mayor "<" y ">".

Received:

Contiene una lista de *tokens* seguida por un punto y coma y un grupo fecha-hora. Cada *token* debe ser un *token* de palabra, *angle-addr*, *addr-spec* o un dominio.

3.3.6 Campos Opcionales

Otros campos distintos a los especificados en las RFC pueden aparecer en los correos. Se deben ceñir a la sintaxis de campos opcionales: un nombre de campo compuesto de caracteres imprimibles US-ASCII (salvo el espacio y el punto) seguido de un texto conforme al *token unstructured*. Los nombres de campo no deben ser idénticos a cualquier otro indicado en las RFC.

3.3.7 Campos de Reenvío

Cuando se reenvía un mensaje se debe añadir un conjunto separado de campos de reenvío. Cada nuevo conjunto de campos de reenvío se agrega por delante al mensaje, esto significa que el conjunto más reciente de campos de reenvío aparecen antes en el mensaje.

Cada uno de los campos de reenvío se corresponde con un campo particular de la sintaxis, por ejemplo el campo ***Resent-Date*** se corresponde con el campo ***Date*** del mensaje original. Cuando se reenvía, los campos de ***Resent-From*** y ***Resent-Date*** deberían incluirse, así mismo sería recomendable que también el campo de ***Resent-Message-Id*** estuviera presente.

La utilidad de estos campos es poder reenviar el mensaje sin alterar los campos originales de ***To***, ***From***, ***Date*** etc.

- ***Resent-Date***: Indica la fecha y hora a la que el mensaje reenviado es despachado por el emisor del reenvío.
- ***Resent-From***: Indica el buzón del emisor del reenvío
- ***Resent-Sender***: no debería ser utilizado si tiene un valor idéntico al de ***Resent-From***
- ***Resent-To***, ***Resent-Cc***, ***Resent-Bcc***: funcionan de manera idéntica a los respectivos campos: ***To***, ***Cc*** y ***Bcc*** salvo que indican al destinatario del mensaje reenviado, no a los destinatarios originales del mensaje (que figuraran en los campos ***To***, ***Cc*** y ***Bcc*** inalterados)
- ***Resent-Msg-ID***: proporciona un identificador único para el mensaje reenviado.

4 ESTRATEGIAS DE ABORDAJE

Se introducen las diversas estrategias para abordar el estudio de un gran corpus de correos electrónicos. Se tratan los tópicos comunes a los diversos objetivos de análisis expuestos en el capítulo segundo de la presente memoria.

En cuanto a la estrategia de persistencia se presentan dos modelos de base de datos (normalizada y des-normalizada tipo en estrella propia de *Data Warehouse*) y uno de grafos puros almacenados en ficheros.

Dentro de las fases de un estudio de estas características se hace especial hincapié en la fase de importación de datos del corpus y en la fase de limpieza del mismo. Durante la importación habrá que generar identificadores únicos y otra información de trazabilidad así como realizar tareas de normalización de la información. La limpieza contemplará: correos duplicados, artificiales y vacíos; de un modo general se tratará de corregir o descartar datos erróneos y campos poco relevantes.

4.1 Determinar el diseño preliminar, objetivos y alcance.

En esta fase se deben delinear los objetivos y el alcance del estudio, tras lo cual se puede aventurar un diseño preliminar de la base de datos en función de los objetivos del estudio. También se puede adelantar parte de la estrategia de importación de los correos.

En esta etapa tendrán cabida las actividades clásicas de la ingeniería del software relacionadas con la captura de requisitos y el análisis de casos de uso que para el presente proyecto se verán en el apartado **[5.2 Diseño del sistema**].

4.1.1 Los objetivos de análisis

Determinar los objetivos del estudio que pueden estar enfocados a:

Análisis de propiedades sociales, clasificación y visualización

Estos objetivos los hemos tratado en el apartado **[2.1 Finalidades de la minería de correo electrónico]**: detección de *spam*, clasificación de correo, análisis de contactos, análisis de las propiedades como red social y visualización de correos.

Análisis estadístico

Se trata de obtener estadísticas de uso que den una información global del conjunto de correos electrónicos, según el alcance, podría ser necesario construir sistemas de *Data Mart* o *Data Warehouse* para obtener informes de grandes volúmenes de datos **[Kimball]**. Para corroborar ciertas hipótesis puede ser necesario contar con análisis estadísticos del conjunto de datos.

Otros análisis

Otros objetivos de análisis pueden tener que ver con recuperar información perdida en los propios procesos del correo electrónico vistos en **[2.1.6 Otras Finalidades]**, como recuperar hilos.

4.1.2 Alcance del estudio

Como en todo proyecto software se requiere definir un alcance para el mismo. Entran en consideración aspectos como:

Carácter estático de los datos

Hay que conocer si se utilizará el sistema sobre el conjunto vivo de datos: se procesarán los correos

desde copias en caliente o se tratará de extracciones nocturnas etc. según las condiciones de los objetivos y la disponibilidad de los datos.

Volumetría del corpus

El tamaño estimado del corpus requerirá dimensionar arquitecturas y elegir enfoques, por ejemplo para objetivos de análisis estadísticos la decisión afectada podría ir desde el uso de tablas indizadas y consultas simples hasta cubos OLAP, etc. Si la arquitectura de la solución va encaminada al uso de grafos puede ser necesario evaluar el particionado o no de los ficheros de grafos en sub-grafos.

Apoyo en otros sistemas

El acceso a otras fuentes de información como directorios LDAP o elementos de información adicional (muchos sistemas envían las convocatorias a reuniones, eventos de calendario, como mensajes de correo electrónico) o enlaces compartidos de repositorios en la nube, todo ello añade nuevas fuentes de información a tener en cuenta.

Privacidad de los correos

Ya se ha citado la privacidad del correo electrónico como un aspecto fundamental a la hora de analizar y explotar en sistemas el correo electrónico. Consideraciones de privacidad, legislación y manuales de buenas prácticas de la corporación en la que se implante el sistema son ejes centrales. Pueden imponerse políticas de desplazamiento de datos, enmascarando cuentas de correo en diccionarios protegidos y la sustitución de los cuerpos de mensajes por texto del tipo *Lorem Ipsum* de la misma longitud. Por desgracia algunos enfoques necesitan contar con el contenido original del mensaje como determinar las personas por su estilo de escritura (firmas al final del correo electrónico, emoticonos, etc).

4.1.3 Definición del diseño preliminar

A la vista de los objetivos y del alcance se plantea un diseño preliminar. Se trata de escoger sobre que arquitectura se va a implementar la solución, si bastará una base de datos relacional, si deberá utilizarse un sistema con especial soporte para análisis de grafos o si la base de datos tendrá que tener estructuras optimizadas para el acceso.

El diseño de las tablas de la base de datos tiene impacto en el tiempo de importación del corpus de correos electrónicos, según el nivel de normalización la inserción será más conveniente y el modelo de datos se mantendrá más cercano al modelo conceptual, pero si el estudio requiere *joins* costosas entre tablas grandes el tiempo de consulta puede crecer en exceso.

Se puede optimizar el espacio y los tiempos de lectura si se almacenan los adjuntos como binarios

en tablas separadas o se guardan solo los enlaces a ficheros. En general cualquier elemento costoso de almacenamiento que no vaya a tomar un papel necesario en el estudio debería evitarse.

Diseño en base de datos normalizada

Se trata de hacer un diseño en torno a la tercera forma normal (*3FN*). Se orienta a los conceptos claves de un análisis de correo electrónico: mensajes, emisores, receptores, etc… aislando las entidades de mayor volumen y eliminando duplicidades para que los cruces entre tablas importantes sea ligero. Por ejemplo guardando en una tabla los mensajes sin repetir y almacenando como relación el hecho del envío del mismo. En la misma línea, se guardan las cabeceras de los correos en otra tabla, ya que para muchos análisis no será necesario acceder al cuerpo.

También se incluye alguna forma de almacenamiento para los grafos de correo electrónico, e hilos de correos si entran dentro del alcance.

A continuación incluimos el diseño de base de datos empleada en el estudio de **[Vadher]**:

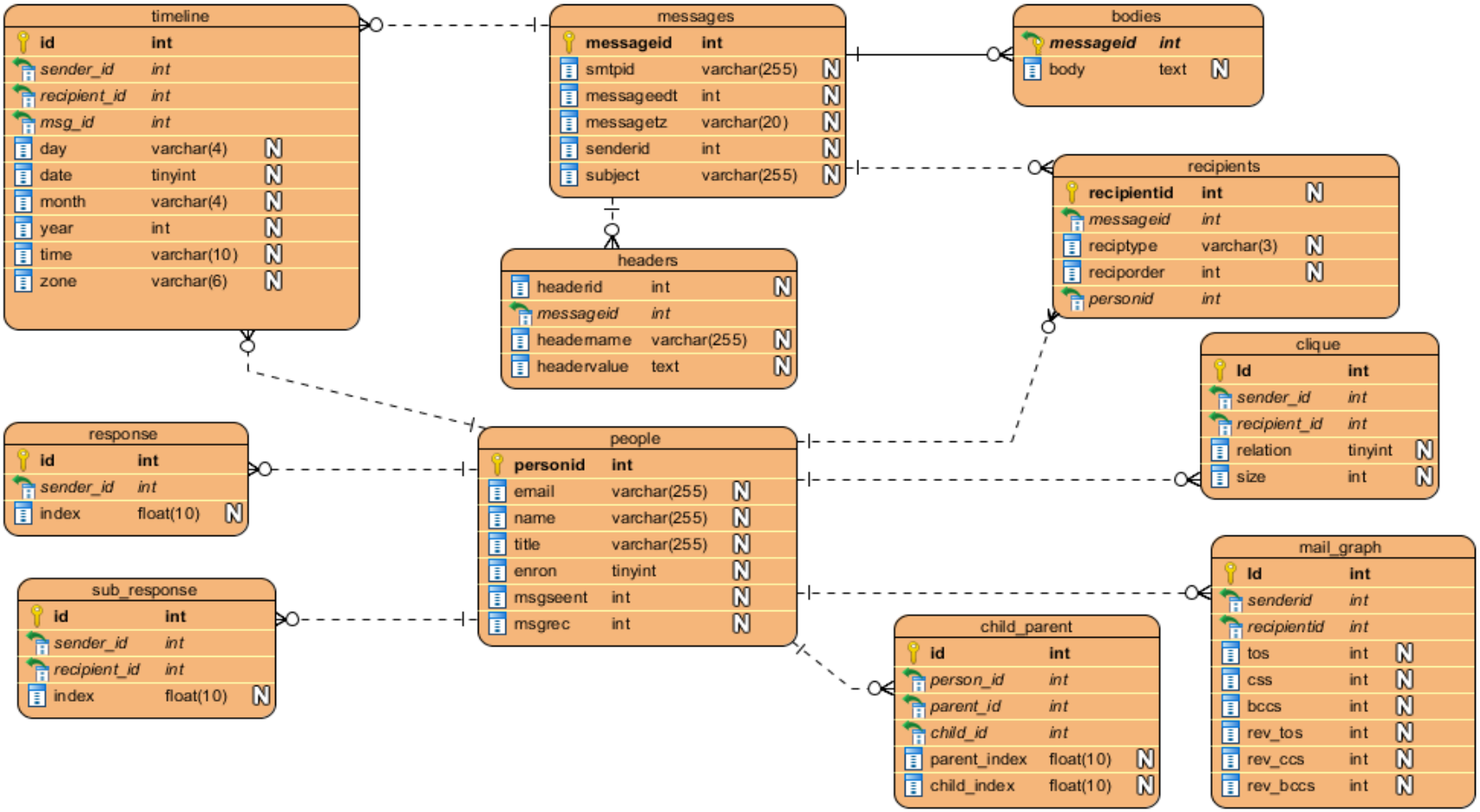


Figura 12: Diseño Normalizado [tomado de Vadher] (continúa…).

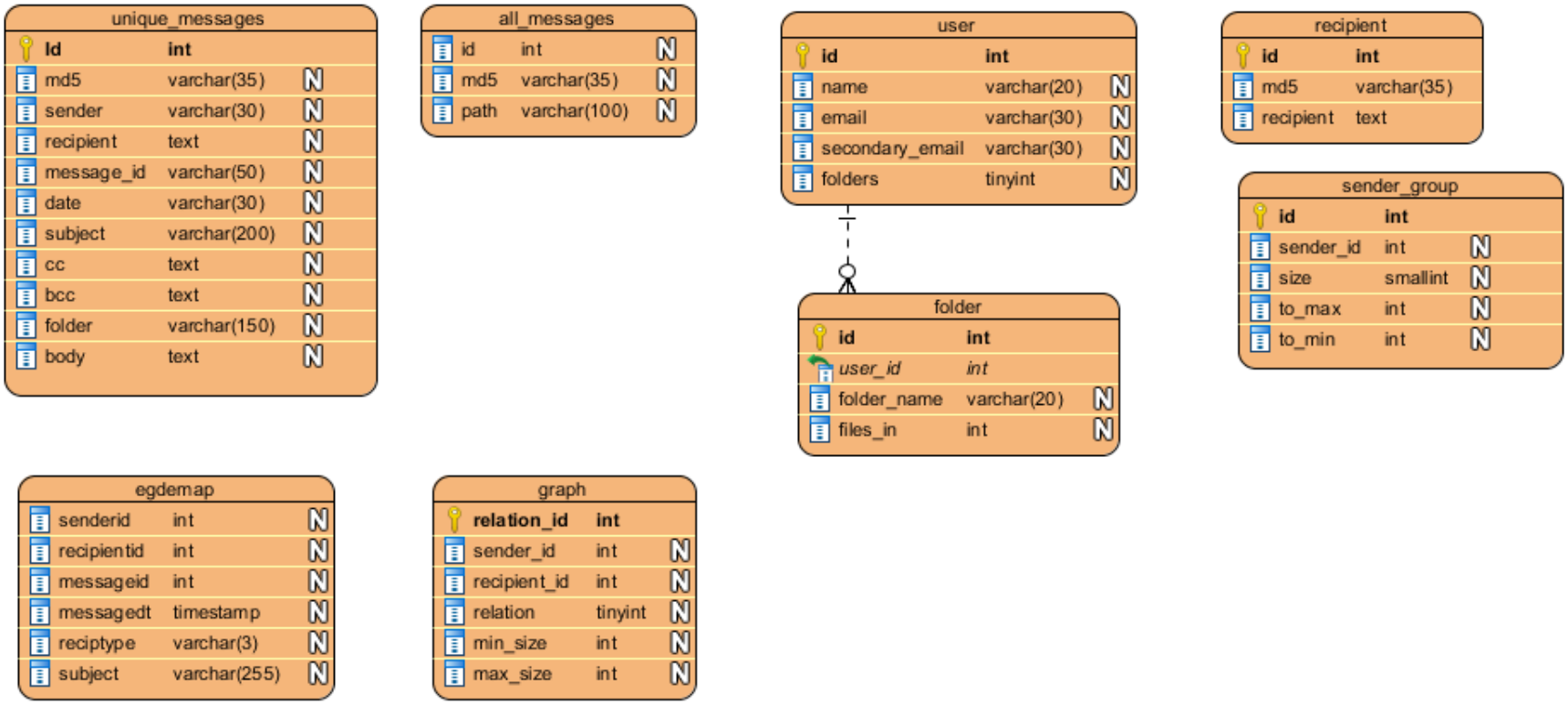


Figura 13: Diseño normalizado tomado de [Vadher] (continuación).

Diseño en base de datos *desnormalizada*

Siguiendo principios de optimización de datos (véase **[Kimball]**), se puede optar por un diseño de base de datos *desnormalizada*. Este tipo de estructura suele ir acompañado de un diseño normalizado a partir del cual se computan datos enlazados en forma de estrella. Requiere más trabajo pero se beneficia de la protección frente a anomalías de inserción que ofrece un diseño normalizado y de la rapidez de cómputo de un sistema *desnormalizado*.

Partiendo de este diseño se pueden utilizar técnicas de cubos OLAP y sistemas de reporte multidimensionales para el análisis de los datos. Aun cuando estemos en el contexto de un análisis de propiedades sociales en forma de grafo, será una herramienta útil para establecer y contrastar hipótesis sobre un conjunto de datos de grandes proporciones.

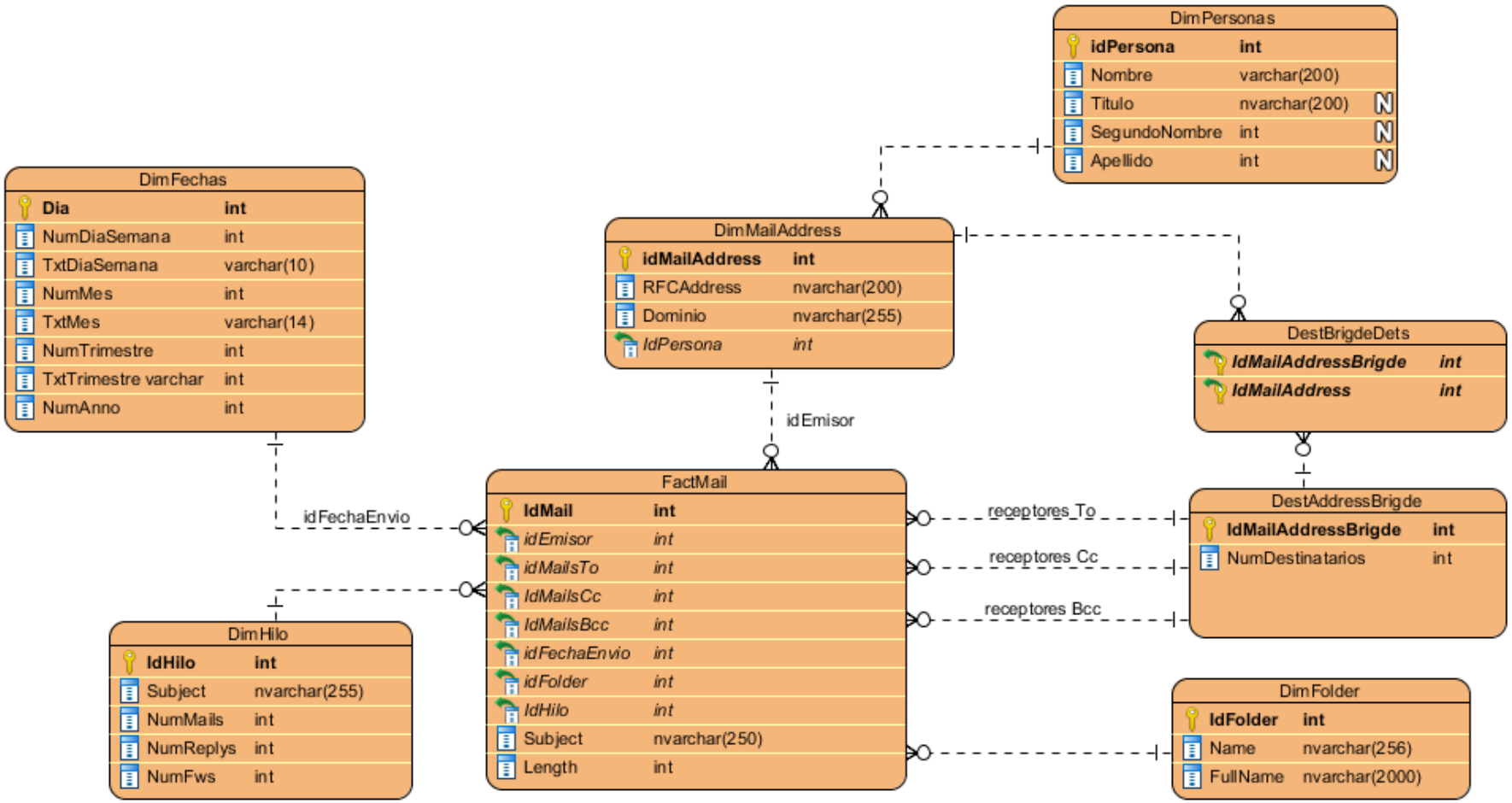


Figura 14: Diagrama de base de datos en estrella.

Diseño en grafos

Las herramientas para gestionar grandes grafos como **[SNAP]** parten de ficheros de carga en los que se introduce la relación entre los nodos.

Herramientas y procesos de carga desde el conjunto de datos origen hacia el formato propio de la librería de grafos. Snap permite cargar y escribir en un fichero de texto plano o en un fichero binario el esqueleto del grafo. El fichero plano será más fácil de procesar pero también tendrá un tamaño mayor, su utilización consumirá más memoria y llevará más tiempo cargarlo y escribirlo

Al leerlo se pueden especificar el tipo de grafo:

- Sin dirigir
    - La modalidad más sencilla de grafo, consta de nodos y aristas que los conectan. Entre dos nodos solo puede haber una arista. Es el caso base para estudiar comunidades donde cada persona es un nodo y la arista indica si hay o no comunicación con los nodos que representan a otras personas.
- Dirigido
    - Cada arista que une dos nodos denota un sentido, la arista está dirigida determinando un nodo origen y un nodo destino que en el contexto de la comunicación se corresponde con los roles de emisor y receptor.
- Multigrafo
    - En esta modalidad se permiten varias aristas entre dos nodos. Podemos representar con ellos múltiples mensajes entre dos nodos. Es interesante remarcar que no todos los sistemas pueden permitir la representación de bucles de un único nodo, algo que en términos de correo electrónico es posible y no muy raro.
- Grafos con atributos
    - Para cualquier modalidad, se pueden emplear etiquetas o atributos tanto en nodos como en aristas.

Si se va a requerir de generación de gráficas o imágenes de mapa de los grafos, hay que elegir alguno de las librerías gráficas que sean compatibles. Para SNAP hay varias opciones:

- GNUPlot: http://gnuplot.sourceforge.net
- GraphViz: http://www.graphviz.org

- Matplotlib: http://matplotlib.org/

4.2 Importación del corpus.

Típicamente se dispondrá de una colección de ficheros de correo en un formato de exportación: ***mbox*** o alguna de sus variantes, fichero PST de ***Outlook***, o algún otro tipo de formato.

Si se parte de conjuntos de datos populares como el de Enron se puede disponer de los ficheros de bases de datos por lo que la fase de importación puede ser innecesaria o con un coste mucho menor, también sucede con otros tipos de interfaces como ***Gmailmeter***, aunque lo más habitual será contar con un flujo de ficheros de correo.

La importación consta de la lectura de los ficheros del corpus y su inserción en las tablas de datos preliminares. Puede optarse por una inserción en una única tabla en bruto sobra la que se trabaje más tarde para rellenar la estructura modelo de tablas del paso anterior o aprovechar esta fase de importación para normalizar algunos datos.

De modo esquemático el proceso de importación de un mensaje de correo electrónico se podría definir como:

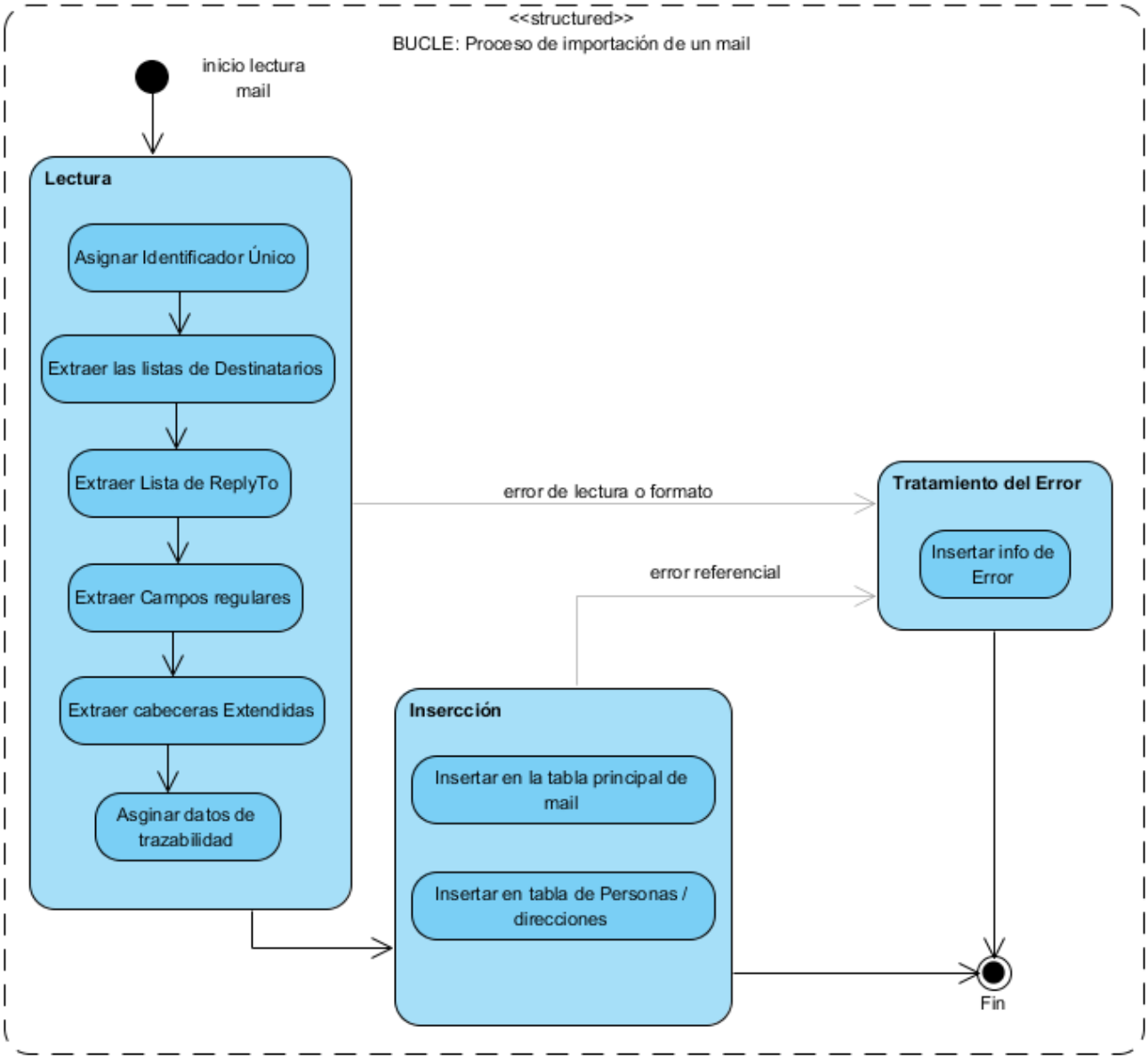


Figura 15: Proceso de importación de un correo electrónico.

4.2.1 Generar Identificadores únicos

Se trata de generar un identificador auto-numérico (*int* o *bigint*) que sirva como clave primaria. Podrá ser usado como “asidero” para las tareas de borrado de duplicados y para referenciar las tablas.

Una alternativa al uso de identificadores enteros auto-numéricos, pueden ser tipos de datos GUID

(*Globally Unique identifiers*). Estos campos son números binarios tales que ninguna otra computadora en el mundo puede generar un valor duplicado. En SQL Server, esos valores se pueden generar mediante el uso de la función *NEWID()* o el uso de *NEWSEQUENTIALID()* que permite reducir el tamaño de página al nivel de las hojas del índice de base de datos.

Utilizar el propio campo ***Message-Id*** del correo electrónico puede ocasionar problemas. En la **[RFC 2822]** se indica que el terminal que genera el mensaje es el responsable de garantizar la unicidad del mismo, pero ese identificador se refiere a una misma versión del mensaje. Si el mismo mensaje aparece en el conjunto de datos, bien porque esté duplicado en la carpeta de borradores, o aparezca en el buzón de la carpeta de enviados y de recibidos, podemos tener errores de *Primary Key* al insertar en la base de datos y tener un proceso de limpieza del conjunto de datos más complejo.

4.2.2 Registrar Información de Trazabilidad

Información adicional como número de fichero, carpeta, usuario en Enron… esta información permite recuperar el fichero original del correo electrónico para contrastar los procesos y poder comprobar errores y corregir problemas.

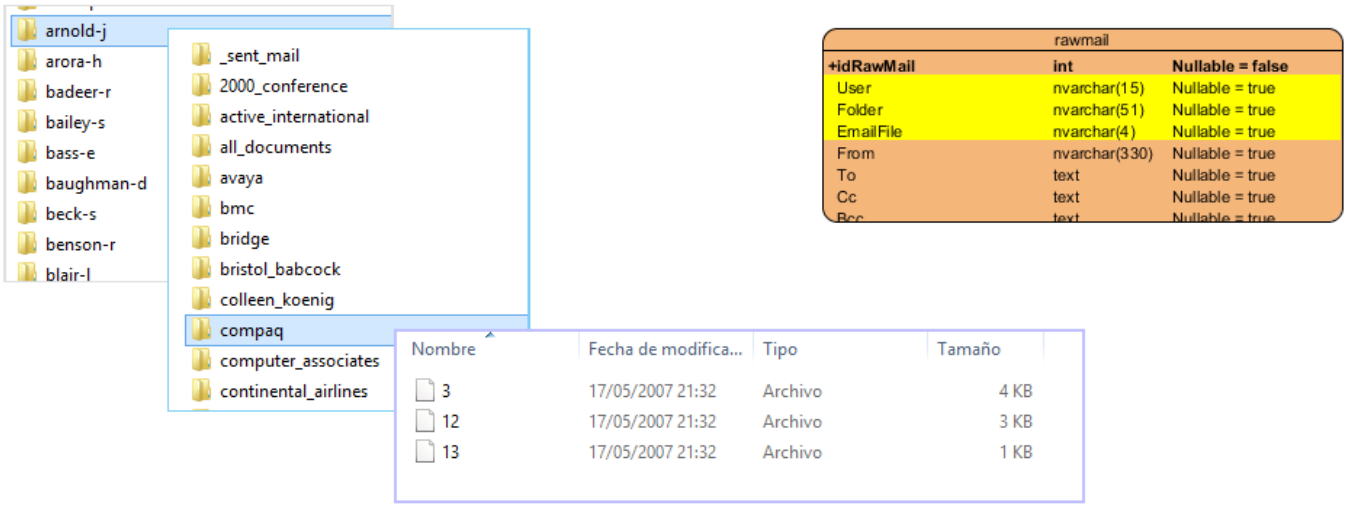


Figura 16: Información de trazabilidad para el caso de Enron.

Agregando esta información al modelo de datos del estudio se facilita seguir casos complejos y detectar el origen de problemas o casos atípicos.

4.2.3 Normalizar Direcciones

En la lectura de ficheros planos, los campos de destinatario aparecerán como listas de direcciones

separadas por comas. Normalizar estos datos en tablas de direcciones y mantener la relación entre el mensaje de correo, el emisor y los destinatarios es mucho más sencillo y eficiente mientras se importa el corpus de un modo secuencial.

4.2.4 Análisis en la etapa de importación

Algunos campos pueden estar incompletos o tener una calidad muy pobre. Es importante determinar si se van a utilizar en el resto del estudio o se van a descartar para las siguientes fases. El principal conjunto de campos que son de vital importancia para el análisis de hilos es el de los campos de identificación, (definidos en **[RFC 2822]**) que permiten reconstruir los hilos de comunicación:

- ***Message-ID***: Contiene un identificador único de mensaje, su existencia y calidad permite utilizar los otros campos para identificar los hilos. Se refiere a la versión de un mensaje particular. La unicidad del identificador del mensaje se garantiza por el terminal que lo genera.
- ***In-Reply-To***: contiene uno o más identificadores únicos de mensaje, separados opcionalmente por espacios en blanco. Se generan cuando se produce un reenvío o una contestación al mensaje original.
- **References**: Se utiliza para identificar un hilo de conversación.

Por desgracia son opcionales y pueden no estar registrados adecuadamente.

4.2.5 Errores de Importación

Cada registro de correo electrónico que provoque algún error debería ser enviado a una tabla de registros erróneos, para analizar la naturaleza del error y continuar con el proceso de importación.

Correos de *spam*, mal formados o mal transmitidos no deberían detener el proceso de importación.

Entre los errores típicos estarán:

- Campos que incumplen reglas obligatorias de formato
  - Direcciones de correo
  - Campos de longitud del correo electrónico
  - Campos de fecha mal formateados
- Campos con tamaños que exceden los límites de las columnas de la base de datos.

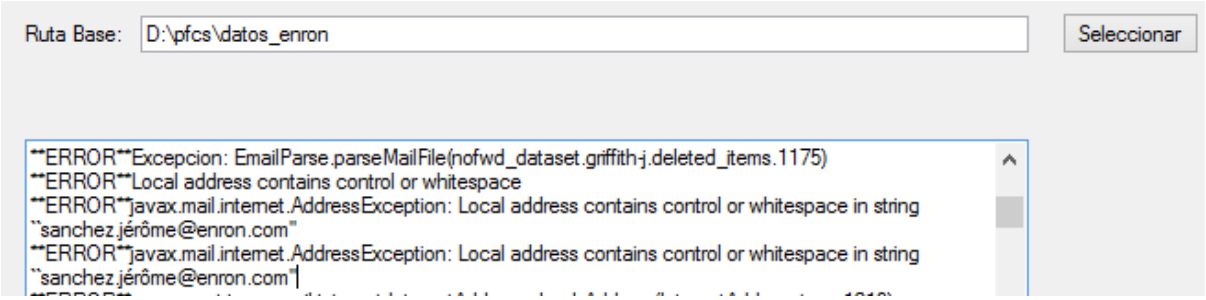


Figura 17: Error al importar Enron: una dirección contiene caracteres especiales.

4.2.6 Estadísticas de velocidad y volumetría

Mantener estadísticas de la velocidad de importación del conjunto de datos puede permitir hacer estimaciones del tiempo que llevará un volumen mayor de datos. Una vez que el sistema esté en producción es conveniente disponer de reportes para verificar que no es necesario redimensionar los elementos de Hardware por el crecimiento de las estructuras de almacenamiento.

4.2.7 Importación diaria

Herramientas de ETL

Si el sistema de análisis se va a implantar en producción bajo unas premisas similares en cuanto a ficheros de origen puede ser conveniente utilizar herramientas especializadas de ETL.

El abanico de este tipo de herramientas va desde scripts sencillos que hagan algunas operaciones hasta tecnologías empresariales pasando por herramientas más ligeras *Open Source*. Algunas opciones podrían ser:

- *Microsoft Integration Services (SSIS)*:
  - La versión de *SQL Server 2005* remplazaba al antiguo DTS, añadiendo mejoras significativas de usabilidad y velocidad aunque al precio de un consumo masivo de memoria. Si se elige como base de datos SQL Server, SSIS ya va incluido en el coste de licencia lo que puede suponer un ahorro frente a otras herramientas de ETL que deberían pagarse a parte.
  - https://msdn.microsoft.com/es-es/library/ms141026%28v=sql.120%29.aspx
- *Scriptella*
  - Es una herramienta *Open Source* programada en java que acepta varios lenguajes

sql, estaría un paso por delante de hacer las ETL con ANT. No dispone de interfaz gráfica, por lo que los procesos se definen en XML.

  - http://scriptella.javaforge.com

- *Power Center*

  - Según una comparativa de *Gatner* (http://www.gartner.com/) especialista en investigación y asesoramiento tecnológico, es la mejor herramienta de ETL. Tiene capacidad para computación en *grid* y permite montar despliegues distribuidos. Incluye un sistema de interfaces gráficas menos atractivo que SSIS, con un coste económico significativamente mayor pero también mejores prestaciones para mover grandes cantidades de datos.

  - https://www.informatica.com/es/products/data-integration/powercenter.html

4.3 Limpieza del conjunto de datos

Una vez se ha importado el corpus por primera vez, será necesario realizar una limpieza previa a la explotación de los datos. Es recomendable limpiar los datos sobre las tablas, donde será más sencillo gestionar la limpieza que eliminarlos en los archivos origen del corpus.

Listamos a continuación las tareas de limpieza, que en cualquier caso dependen de la naturaleza del estudio:

4.3.1 Correos electrónicos duplicados

El caso más obvio de correos repetidos es el del mensaje que se mantiene en la carpeta de enviados del emisor y en la carpeta de recibidos del destinatario. Existen otros muchos contextos de uso cotidiano de correo electrónico en el que se almacena el mismo mensaje varias veces: reglas de correo, múltiples direcciones para una misma persona: listas de distribución de departamento y grupo de trabajo.

Según el objeto de estudio puede ser o no relevante quedarnos con solo una copia. Se trata de quedarnos con el hecho de la comunicación: un emisor ha enviado un mensaje a uno o más receptores en un momento determinado.

En otras ocasiones puede ser interesante constatar el hecho de personas que han eliminado de su buzón un mensaje en el que sabemos eran destinatarios, podemos o no inferir que el correo nunca llegó a la cuenta aunque es más verosímil que se eliminara y se vaciara la carpeta de eliminados si en

el buzón del emisor no existe un devolución del correo.

En el caso de Enron **[Shetty]** prescinde de los mensajes de las carpetas *discussion_thread* y *all_documents* por estar repetidos en otras carpetas y buzones.

Para determinar qué condiciones se aplican a la hora de determinar si un correo electrónico está duplicado, algunos autores como **[Yuan y Harnly]** sugieren tomar como mensajes duplicados aquellos que tienen los mismos valores para:

- *Subject*
- *Body*
- *From*
- *To, Cc, Bcc*
- Marca de Tiempo (*Timestamp*):
    - Conviene tener un umbral de al menos un día dado que los sistemas de correo pueden desplazar la marca de tiempo para el mismo correo entre la carpeta del emisor y la carpeta del receptor.

4.3.2 Datos erróneos

Algunos campos pueden contener valores erróneos. Listamos algunos de los encontrados en el corpus de Enron.

Fechas erróneas

Errores de configuración de equipos o de transmisión hacen que, aunque raros, existan correos con fechas futuras y pasadas. Se pueden eliminar esos datos en base a un criterio temporal estricto, entre el 1999 y el 2001 abarcan los correos electrónicos de Enron, o bien tratarlos como datos atípicos y eliminarlos de un modo estadístico como *n* veces la desviación estándar.

Para el presente trabajo se ha optado por este segundo enfoque.

```
SELECT
    [idRawMail],
    [User],
    [Folder],
    [EmailFile],
    [From],
    [Date],
    [MessageId],
    [Subject],
    [ReplyTo],
    [Recipients]
FROM
    rawmail
WHERE
    YEAR([Date]) NOT BETWEEN 1999 AND 2002
ORDER BY
    [Date]
```

	idRawMail	User	Folder	EmailFile	From	Date
1	1080	arnold-j	all_documents	141	john.arnold@enron.com	1980-01-01 01:00:00.000
2	1978	arnold-j	discussion_threads	21	john.arnold@enron.com	1980-01-01 01:00:00.000
3	1979	arnold-j	discussion_threads	26	john.arnold@enron.com	1980-01-01 01:00:00.000
4	2031	arnold-j	discussion_threads	66	john.arnold@enron.com	1980-01-01 01:00:00.000
5	4233	bass-e	all_documents	10	eric.bass@enron.com	1980-01-01 01:00:00.000
6	4296	bass-e	all_documents	9	eric.bass@enron.com	1980-01-01 01:00:00.000
7	4666	bass-e	discussion_threads	10	eric.bass@enron.com	1980-01-01 01:00:00.000

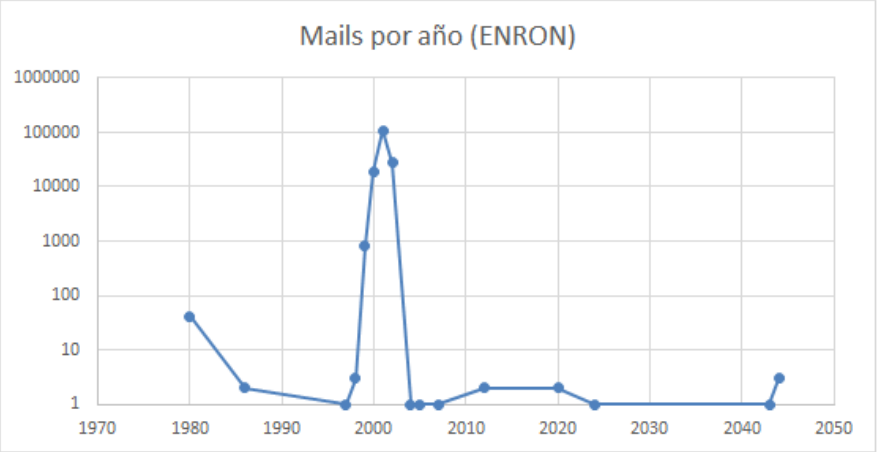


Figura 18: Limpieza de Fechas incorrectas en el corpus de Enron.

Representando los mensajes de correo por año en escala logarítmica en base 10, para el corpus de Enron vemos que algunos años acumulan correos con fechas incorrectas.

Direcciones de correo

Algunas direcciones en campos de destinatario pueden tener formato erróneo según las reglas de las RFC, nombres de destinatarios que no se han mapeado correctamente a una dirección real de correo. Para el caso de Enron no se ha encontrado ningún caso en el que la dirección estuviera mal formada.

Destinatarios no divulgados

Las normas de etiqueta y estilo corporativo dictan que los mensajes enviados a un gran número de personas incluyan a los destinatarios en ***Bcc*** para no exponer las cuentas de correo en el campo ***To***. La mayoría de clientes de correo recomiendan introducir al menos una dirección de destinatario en el campo ***To***, para que los MTA no tilden de *spam* el correo por no tener dirección ***To***. En estos casos se introduce una dirección del estilo "*undisclosed recipients*" (destinatarios no divulgados) en el ***To***.

Para el caso de Enron, existen varias cuentas de este estilo undisclosed@enron.com. Algunos autores como **[Shetty]** sugieren asociar las direcciones mal formadas en el campo ***To*** a esta dirección.

4.3.3 Correos electrónicos artificiales

Algunas carpetas de las personas implicadas contienen mensajes autogenerados por sus clientes de correo o por el sistema corporativo de correo.

En el conjunto de datos de Enron un ejemplo que se cita en **[Klimt]** son las carpetas "*discussion thread*" por no ser correos originales de los empleados de Enron.

A continuación presentamos una gráfica del número de mensajes en la carpeta *discussion_threads* por buzón de usuario en el conjunto de datos.

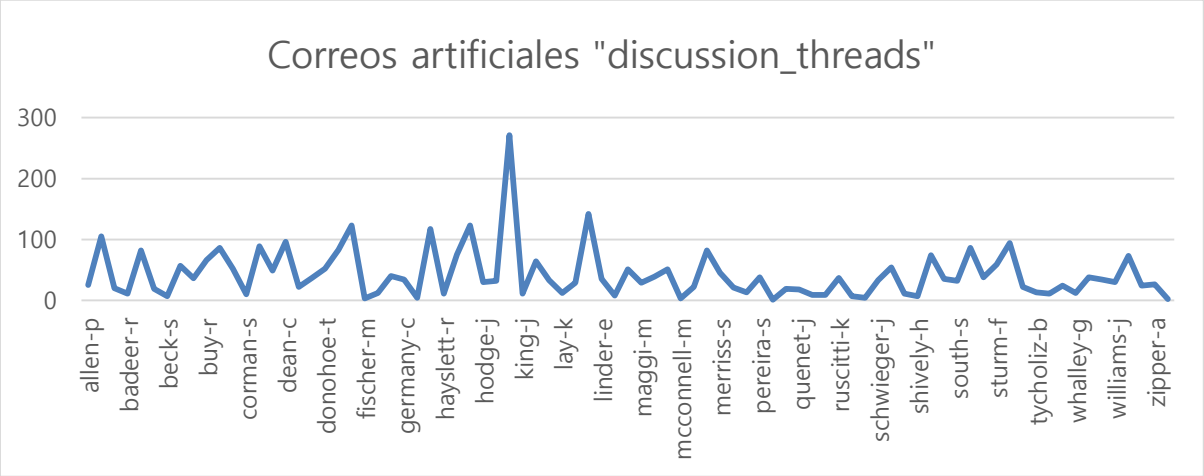


Figura 19: Correos artificiales en la carpeta discussion_threads en el corpus de Enron.

Hay un total de 3.543 mensajes en carpetas con ese nombre, que sobre el total de correos (157.496) no supone más de un 2,25%.

4.3.4 Correos electrónicos vacíos

Los mensajes sin cuerpo, recordemos que la **[RFC 822]** indica que el cuerpo es opcional, son suprimidos de sus estudios por algunos autores **[Shetty]** que requieren del análisis del cuerpo del mensaje. Depende del estudio concreto pero si es una decisión a tener cuenta.

Para el caso de Enron encontramos 15 correos electrónicos con el cuerpo del mensaje vacío:

Resultados | Mensajes

	idRawMail	User	Folder	EmailFile	From	To	Cc	Bcc	Body
1	16196	carson-m	deleted_items	206	feedback@intcx.com	powerindex@list.intcx.com			
2	16204	carson-m	deleted_items	214	feedback@intcx.com	powerindex@list.intcx.com			
3	41464	griffith-j	deleted_items	922	feedback@intcx.com	gasindex@list.intcx.com			
4	41475	griffith-j	deleted_items	937	feedback@intcx.com	gasindex@list.intcx.com			
5	41476	griffith-j	deleted_items	938	feedback@intcx.com	powerindex@list.intcx.com			
6	85926	martin-t	inbox	416	feedback@intcx.com	gasindex@list.intcx.com			
7	85941	martin-t	inbox	430	feedback@intcx.com	gasindex@list.intcx.com			
8	86508	may-l	inbox	203	feedback@intcx.com	gasindex@list.intcx.com			
9	86515	may-l	inbox	212	feedback@intcx.com	gasindex@list.intcx.com			
10	86554	may-l	inbox	262	feedback@intcx.com	gasindex@list.intcx.com			
11	86580	may-l	inbox	291	feedback@intcx.com	gasindex@list.intcx.com			
12	86584	may-l	inbox	296	feedback@intcx.com	gasindex@list.intcx.com			
13	86585	may-l	inbox	297	feedback@intcx.com	powerindex@list.intcx.com			
14	105975	ring-a	inbox	52	feedback@intcx.com	gasindex@list.intcx.com			
15	105978	ring-a	inbox	56	feedback@intcx.com	gasindex@list.intcx.com			

Figura 20: Mails con el cuerpo vacío en el corpus de Enron.

4.3.5 Direcciones y/o buzones distintos para la misma persona

Un caso presente en el corpus de Enron es el de buzones distintos para una misma persona. En este caso resulta de interés unificar ambos buzones.

Algunas estrategias para agrupar direcciones son:

Unificar por nombres y apellidos

Está sujeto a errores provocados por la coincidencia de la inicial del nombre y la mezcla de apellidos. Depende de la política de asignación de cuentas de correo de la corporación y finalmente si en el conjunto de datos se pueden mezclar cuentas de personales, estás pueden no contener ningún componente del nombre y apellidos de la persona. No obstante pueden ser un buen punto de partida si el corpus es corporativo.

Características en el lenguaje

Como se mencionó en [**2.1.3 Análisis de contactos**] una aproximación empleada para unificar los alias de correo de una misma persona es buscar características en el lenguaje empleado en el cuerpo del correo electrónico, además de firmas de correo y emoticonos usados. Puede suponer un estudio a parte y el éxito depende en gran medida de la disponibilidad de un conjunto suficiente de muestras

para catalogar bien las personas y sus buzones.

4.3.6 Campos obsoletos o poco relevantes

Algunos campos no estarán presentes o su valor en el conjunto de datos será irrelevante.

Por ejemplo en el conjunto de datos de Enron, el campo *Reply-To* siempre es igual al campo *From*, por lo que no aporta ninguna información.

Ocurre algo similar cuando la variabilidad en los valores es muy baja, aunque en este caso depende más del objeto de estudio. Un ejemplo en los datos de Enron es ***content-transfer-encoding***. Si el corpus incluye adjuntos y solo necesitamos su nombre de fichero, su tamaño en octetos y puede que un *checksum*, podemos descartar los adjuntos y ahorrar algo de espacio en la base de datos.

Algunos campos proporcionados por las librerías de lectura de los buzones de correo (javax.mail por ejemplo) estarán relacionados con el uso "en vivo" del correo: *flags* de leído, urgente etc. En general deben ser rechazados porque no reflejan el estado del correo leído sino las condiciones de la aplicación "virtual" en la que se han cargado.

4.3.7 Herramientas para la limpieza de datos

Para este proyecto la limpieza se ha hecho con scripts SQL desde la propia base de datos, aunque existen otras herramientas como DQS (*Data Quality Services*) incluida en *MS SQL Server 2012* que permiten una limpieza del contenido de los datos.

Este tipo de herramientas se relacionan con *Sistemas de Conocimiento* y se usan para tareas de corrección, enriquecimiento, estandarización y eliminación de duplicados.

Con el deterioro en la calidad que introducen los correos de *spam*, y las interpretaciones de cabeceras propias de cada fabricante, la limpieza de los datos podrá ser más o menos preventiva y consumir un esfuerzo considerable por lo que en este punto se deberá alcanzar una solución de compromiso.

4.4 Tareas tras la primera importación

4.4.1 Evaluar las cabeceras extendidas

Según la época y las aplicaciones de correo el conjunto de datos puede contener importantes cabeceras extendidas para el estudio. Conviene analizar que cabeceras extendidas están presentes y

su frecuencia. A modo de ejemplo citamos algunas presentes en el conjunto de datos de Enron:

X-To, ***X-From***, ***X-Bcc***, ***X-cc***: Estas cabeceras de destinatario y origen están extendidas para indicar información de envío fuera del estándar **[RFC 822]**. Contienen el nombre de la persona en la libreta de direcciones del emisor, a veces un alias o un pseudónimo de conexión, en cualquier caso suele ser una información adicional y pareja a la dirección estándar de correo electrónico.

4.4.2 Afinar los tipos de datos y sus longitudes

Tamaño global:

Ajustar el tamaño del corpus para cubrir los objetivos de estudio, es necesario para optimizar los tiempos de procesamiento y los análisis, reduce además el tamaño de copias de seguridad y en general simplificará el procesamiento de los datos, reduciendo los tipos de datos y el tamaño completo de la base de datos o los ficheros empleados. Un método muy efectivo para compactar el conjunto de datos es eliminar los correos electrónicos duplicados.

Tipos de datos y tamaños de columnas:

Algunas circunstancias pueden hacer especialmente deseable reducir el tamaño de las columnas al máximo permitido para la clave primaria o clave única. Si un campo está almacenado como BLOB o *TEXT* no solo no podrán formar parte de un índice sino que algunas facilidades proporcionadas por el sistema gestor de bases de datos no serán utilizables directamente como obtener valores distintos con *DISTINCT*, o clausulas *GROUP BY*.

En el caso de trabajar directamente con ficheros sin utilizar base de datos relacional, determinadas funciones de lectura pueden ser adecuadas a ficheros grandes o pequeños.

5 ANÁLISIS Y DISEÑO

Se expone el análisis del sistema *software* sobre el que versa el presente proyecto fin de carrera.

En la primera parte del capítulo se presentarán los casos de uso, una descripción detallada y los requisitos de usuario de capacidad y restricción.

En la segunda parte de diseño se presenta cada componente modular del sistema así como el diseño de la base de datos y el diseño del subsistema de informes.

5.1 Análisis del sistema

En esta fase se define el problema al que se quiere dar solución, atendiendo al dominio natural de la aplicación y a la funcionalidad requerida.

Se trabaja desde la perspectiva del usuario del sistema que será a priori un analista técnico o un equipo de analistas dispuestos a explotar uno o más corpus de correos electrónicos obtenidos de distintas fuentes.

Se analizan los casos de uso del sistema, para analizar las interacciones del usuario y el sistema delimitando el alcance del mismo. En un paso posterior esos casos proporcionan el origen de discusión de los requisitos de usuario de capacidad y restricción.

Los requisitos de usuario de capacidad definen aquello que el cliente desea que el sistema haga en el dominio del problema.

Los requisitos de usuario de restricción indican la manera que tendrá el sistema de solucionar las necesidades del cliente, la forma de interacción con el mismo y las restricciones en el uso.

5.1.1 Casos de Uso

A continuación se presenta el diagrama de casos de uso.

Figura 21: Casos de uso (general).

La generación de los reportes de informe se puede explosionar a su vez en un buen número de sub-casos, los representaremos en el siguiente diagrama:

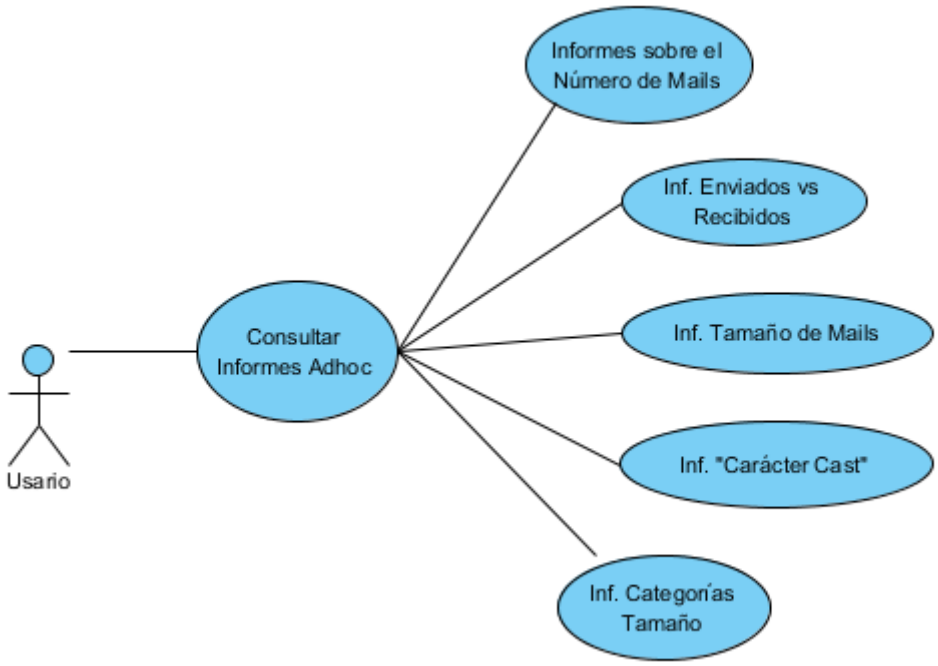


Figura 22: Casos de uso (Consultar Informes Adhoc).

Cada familia de informes se divide en informes concretos que permite consultar la información desde diversas perspectivas.

Algunos casos como *Correos Enviados vs Recibidos* se deben aplicar siempre por agrupaciones de direcciones, esto es porque al calcular el número de correos enviados y recibidos siempre hay que aplicarlo a un grupo de direcciones. En caso contrario el número de mensajes recibidos ha de ser al menos igual al de enviados (dado que el hecho de un correo electrónico implica un emisor y uno o más receptores).

Las perspectivas más comunes que se buscará ofrecer serán: por año, meses, día, día de la semana, horas y en algunos casos franjas horarias.

Resumimos en la siguiente tabla las perspectivas de cada tipo de informe:

	Número de Correos	Enviados vs Recibidos	Tamaño de correo	Carácter *Cast*	Categorías de Tamaño
Anual	Por Años	Por Agrupación y Año	Por Agrupación y Año	Por Agrupación y Año	Por Agrupación y Año
Meses	Por Meses	Por Agrupación y Mes	Por Agrupación y Meses	Por Agrupación y Mes	Por Agrupación y Mes
Día		Por Agrupación y Día	Por Agrupación y Día	Por Agrupación y Día	Por Agrupación y Día
Día de la Semana	Por Día de la Semana	Por Agrupación y Día de la semana	Por Agrupación y Día de la semana	Por Agrupación y Día de la semana	Por Agrupación y Día de la semana
Franja Horaria	Por Franja Horaria	Por Agrupación y medias horas			
Agrupación	Por Agrupación				Por Agrupación
Otros				Por categoría "cast"	

Tabla 2: Perspectivas para cada informe.

5.1.2 Descripción detallada.

Se requiere un sistema de análisis estadístico de envío y recepción de correo electrónico que sirva como base para futuros estudios más elaborados.

- El sistema debe ser capaz de importar corpus de correos electrónicos desde varios orígenes: ficheros de texto planos distribuidos en ficheros de texto plano y carpetas, archivos de correo **PST**, y en formato de fichero único ***mbox***.
- El sistema debe ofrecer diversas funcionalidades de configuración para definir ciertas categorías de agrupación, carácter *cast* y categorías de tamaño para los análisis.
- También se debe poder eliminar correos con fechas incorrectas que puedan suponer ruido para el análisis.
- La salida de los análisis del sistema serán una aplicación de informes que permita consultar las distintas magnitudes bajo diversas perspectivas. La otra salida será un archivo de *OOXml* (formato reconocido por *Office 2007, Libre Office*…) que tendrá un resumen de las estadísticas del corpus.

Este trabajo no pretende abordar todas las grandes cuestiones relativas a la minería de correo electrónico que hemos apuntado en el capítulo segundo: **[Estado de la Cuestión]**, sino más bien cubrir los pasos básicos para abordar cualquier trabajo complejo en esa área.

5.1.3 Requisitos de Usuario

La especificación de requisitos de usuario es una tarea habitual en el proceso de análisis de un sistema. Se trata de identificar, clasificar y catalogar requisitos para el sistema a desarrollar.

El formato con el que registran en la presiente memoria incluye los campos:

- **Identificador**: un código para hacer el seguimiento del requisito, a la derecha presentamos un epígrafe de referencia para identificar también al requisito.
- **Prioridad**: permite ajustar la planificación en la fase de desarrollo.
- **Fuente**: el origen del requisito que en este caso será el tutor o el alumno.
- **Necesidad**: Los requisitos esenciales se deben incorporar en el sistema, aquellos deseables u opcionales pueden ser negociables.
- **Descripción**: Se describe de manera clara y concisa el requisito.
- **Claridad**: Se valora la simpleza de la descripción del requisito.
- **Verificabilidad**: Se valora la capacidad de comprobar si el sistema cumple el requisito.

Requisitos de usuario de capacidad

Identificador: RUC-001			*Herramienta de análisis estadístico*
Prioridad:	**Alta** *Media Baja*	**Fuente:**	Tutor
Necesidad:	**Esencial** *Deseable Opcional*		
Descripción:	Para la realización del proyecto será necesario ofrecer una herramienta de análisis estadístico de un corpus de correo electrónico.		
Estabilidad:	Estable		
Claridad:	*Alta* **Media** *Baja*	**Verificabilidad:**	**Alta** *Media Baja*

Tabla 3: Req. de usuario de capacidad 001, Herramienta de análisis estadístico.

Identificador: RUC-002			*Múltiples Estudios*
Prioridad:	*Alta* **Media** *Baja*	**Fuente:**	Tutor
Necesidad:	*Esencial* **Deseable** *Opcional*		
Descripción:	El sistema debe mantener almacenados los datos de uno o más estudios sobre corpus de correo electrónico. De este modo la plataforma de análisis podrá tener distintos corpus no siendo necesario instalar distintas instancias para trabajar en distintos estudios.		
Estabilidad:	Estable		
Claridad:	**Alta** *Media Baja*	**Verificabilidad:**	**Alta** *Media Baja*

Tabla 4: Req. de usuario de capacidad 002, Múltiples Estudios.

Identificador: RUC-003			*Importar, formatos de entrada*
Prioridad:	**Alta** *Media Baja*	**Fuente:**	Tutor
Necesidad:	**Esencial** *Deseable Opcional*		
Descripción:	El sistema debe aceptar al menos tres formatos de almacenamiento de correos electrónicos como elemento de entrada para importar datos.		
Estabilidad:	Estable		
Claridad:	**Alta** *Media Baja*	**Verificabilidad:**	*Alta* **Media** *Baja*

Tabla 5: Req. de usuario de capacidad 003, Importar y formatos de entrada.

Identificador: RUC-004			*Agrupaciones*
Prioridad:	*Alta* **Media** *Baja*	**Fuente:**	Alumno
Necesidad:	*Esencial* **Deseable** *Opcional*		
Descripción:	El sistema debe permitir que se introduzcan grupos de direcciones de correo en base al dominio de la dirección para agrupar las magnitudes de estudio de los mismos.		
Estabilidad:	Estable		
Claridad:	**Alta** *Media Baja*	**Verificabilidad:**	**Alta** *Media Baja*

Tabla 6: Req. de usuario de capacidad 004, Agrupaciones.

Identificador: RUC-005			*Configurar Periodos*
Prioridad:	*Alta* **Media** *Baja*	**Fuente:**	Tutor
Necesidad:	*Esencial* **Deseable** *Opcional*		
Descripción:	El sistema debe contar con opciones de configuración para definir algunos periodos de tiempo. En concreto: • Debe poder definir los tramos horarios, incluyendo por defecto la configuración madrugada, mañana, mediodía, tarde y noche. • Debe poder definir los trimestres (*Quarters*) para que se puedan adecuar mejor a un objeto de estudio concreto. Por defecto se incluirán los valores: ○ Q1: Enero a Marzo ○ Q2: Abril a Junio ○ Q3: Julio a Septiembre ○ Q4: Octubre a Diciembre		
Estabilidad:	Estable		
Claridad:	**Alta** *Media* *Baja*	**Verificabilidad:**	**Alta** *Media* *Baja*

Tabla 7: Req. de usuario de capacidad 005, Configurar Periodos.

Identificador: RUC-006			*Limpiar datos*
Prioridad:	**Alta** *Media Baja*	**Fuente:**	Tutor
Necesidad:	**Esencial** *Deseable Opcional*		
Descripción:	El sistema debe incluir herramientas para limpiar el conjunto de datos de: • fechas incorrectas o sospechosas por estar alejadas del resto de fechas de correo • correos duplicados según los criterios de contenido del cuerpo del mensaje, emisor y receptores.		
Estabilidad:	Estable		
Claridad:	**Alta** *Media Baja*	**Verificabilidad:**	**Alta** *Media Baja*

Tabla 8: Req. de usuario de capacidad 006, Limpiar Datos.

Identificador: RUC-007			*Informe resumen OOXml*
Prioridad:	**Alta** *Media Baja*	**Fuente:**	Tutor
Necesidad:	*Esencial* **Deseable** *Opcional*		
Descripción:	El sistema debe generar un informen resumen en formato de hoja de cálculo *Office Open* XML.		
Estabilidad:	Estable		
Claridad:	**Alta** *Media Baja*	**Verificabilidad:**	**Alta** *Media Baja*

Tabla 9: Req. de usuario de capacidad 007, Informe resumen OOXml.

Identificador: RUC-008			*Informes adhoc*
Prioridad:	**Alta** *Media Baja*	**Fuente:**	Alumno
Necesidad:	*Esencial* **Deseable** *Opcional*		
Descripción:	El sistema debe ofrecer una interfaz de consulta que permita obtener algunos análisis estadísticos del conjunto de datos. En concreto debe ofrecer informes de las siguientes magnitudes: • Número de correos electrónicos • Número de correos electrónicos enviados vs recibidos • Carácter *cast* de las comunicaciones (clasificación de correos según el número de destinatarios) • Tamaño en Kilo *bytes* de los mensajes • Clasificación en categorías de tamaño		
Estabilidad:	Estable		
Claridad:	**Alta** *Media Baja*	**Verificabilidad:**	**Alta** *Media Baja*

Tabla 10: Req. de usuario de capacidad 008, Informes adhoc.

Requisitos de usuario de restricción

Identificador: RUR-001			*Windows .NET MSSQL*
Prioridad:	*Alta* **Media** *Baja*	**Fuente:**	Alumno
Necesidad:	*Esencial* **Deseable** *Opcional*		
Descripción:	El sistema se desarrollará bajo plataforma *Windows*, tecnología .NET y con *SQL Server 2008* o superior como sistema gestor de bases de datos.		
Estabilidad:	Estable		
Claridad:	**Alta** *Media Baja*	**Verificabilidad:**	**Alta** *Media Baja*

Tabla 11: Req. de usuario de restricción 001: Windows .NET MSSQL.

Identificador: RUR-002			*MS Outlook PST*
Prioridad:	**Alta** *Media Baja*	**Fuente:**	Alumno
Necesidad:	*Esencial* **Deseable** *Opcional*		
Descripción:	El sistema utilizará librerías de interoperación de *MS Office* para la lectura de ficheros PST de *Outlook*.		
Estabilidad:	Estable		
Claridad:	*Alta* **Media** *Baja*	**Verificabilidad:**	**Alta** *Media Baja*

Tabla 12: Req. de usuario de restricción 002: MS Outlook PST.

Identificador: RUR-003			*Info. Progreso*
Prioridad:	**Alta** *Media Baja*	**Fuente:**	Tutor
Necesidad:	*Esencial* **Deseable** *Opcional*		
Descripción:	Durante el proceso de carga de los correos electrónicos de un corpus debe presentarse información de progreso y del tiempo empleado al usuario.		
Estabilidad:	Estable		
Claridad:	**Alta** *Media Baja*	**Verificabilidad:**	**Alta** *Media Baja*

Tabla 13: Req. de usuario de restricción 003: Info. Progreso.

Identificador: RUR-004			*Importación con parada y reanudar*
Prioridad:	**Alta** *Media Baja*	**Fuente:**	Tutor
Necesidad:	*Esencial* **Deseable** *Opcional*		
Descripción:	Los procesos de importación han de poder dejarse en pausa y ser reanudados mientras no se salga de la pantalla de importación. La salida de la pantalla de importación ha de requerir en cualquier caso una confirmación por parte del usuario.		
Estabilidad:	Estable		
Claridad:	*Alta* **Media** *Baja*	**Verificabilidad:**	**Alta** *Media Baja*

Tabla 14: Req. de usuario de restricción 004: Importación con parada y reanudar.

Identificador: RUR-005			*Req. Importar y Configurar*
Prioridad:	**Alta** *Media Baja*	**Fuente:**	Alumno
Necesidad:	*Esencial* **Deseable** *Opcional*		
Descripción:	Se requerirá al usuario realizar al menos una importación de correos electrónicos y una configuración del estudio como requisito para acceder a los informes de ese estudio.		
Estabilidad:	Estable		
Claridad:	**Alta** *Media Baja*	**Verificabilidad:**	**Alta** *Media Baja*

Tabla 15: Req. de usuario de restricción 005: Req. Importar y Configurar.

Identificador: RUR-006			*Gráficas en informes*
Prioridad:	*Alta Media* **Baja**	**Fuente:**	Alumno
Necesidad:	*Esencial* **Deseable** *Opcional*		
Descripción:	Todos los informes deberán ir acompañados de gráficas para facilitar una visión panorámica al usuario.		
Estabilidad:	Estable		
Claridad:	*Alta* **Media** *Baja*	**Verificabilidad:**	**Alta** *Media Baja*

Tabla 16: Req. de usuario de restricción 006: Gráficas en Informes.

5.2 Diseño del sistema

El diseño consta de una aplicación con tres subsistemas fácilmente identificables: *Gestión de Estudios*, *Importación* e *Informes*. Se emplea una base de datos *SQL Server* para mantener los datos de configuración general de la aplicación y una base de datos por cada estudio de corpus de correo electrónico que se realice.

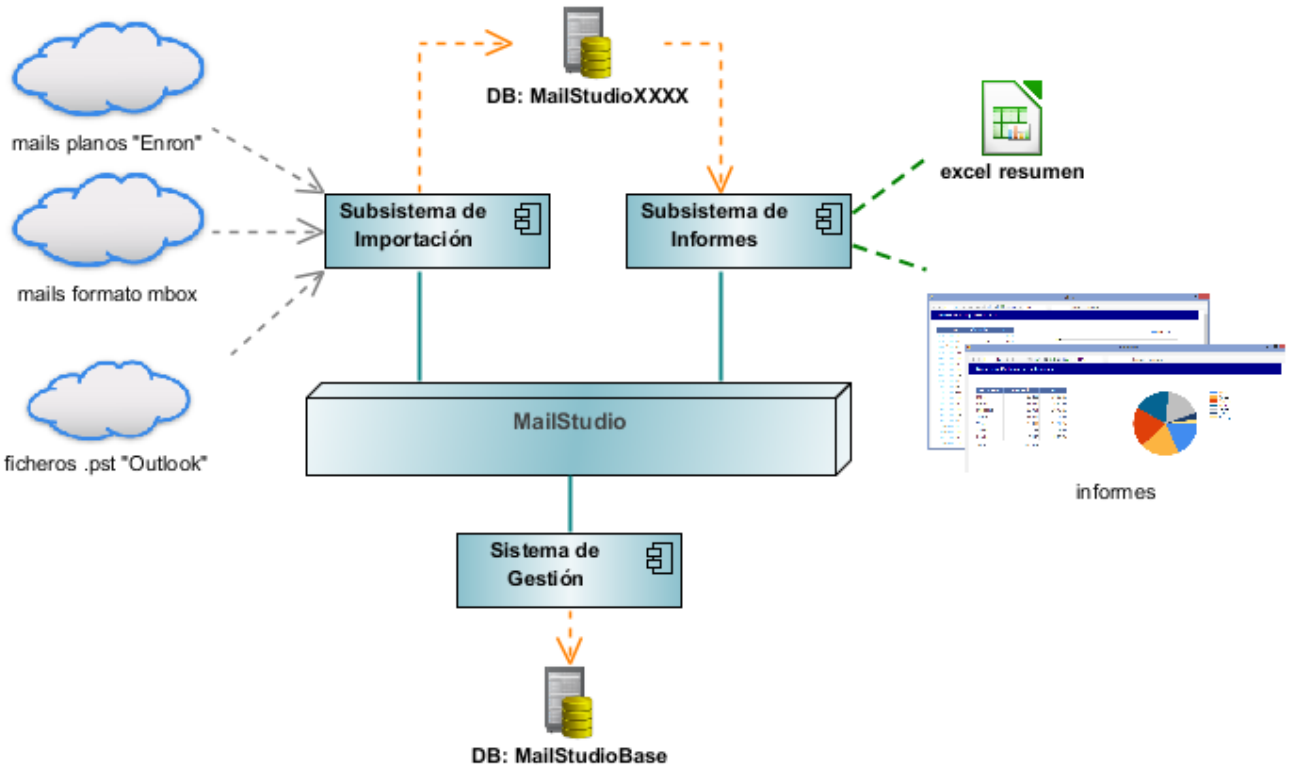


Figura 23: Diseño general del sistema.

5.2.1 Subsistema de Importación.

El subsistema de importación comprende la lectura del fichero o ficheros que contiene el corpus de correos electrónicos. Por simplicidad en el sistema se maneja la lógica externa de importación (comenzar, parar, reanudar, notificar el grado de avance de la tarea, etc…) en la parte de la interfaz de usuario bajo cada uno de los paneles de control de cada tipo de importación.

Para un sistema más grande dicha lógica podría desplazarse a clases controladoras más sofisticadas quizás siguiendo un patrón de tipo Factoría Abstracta.

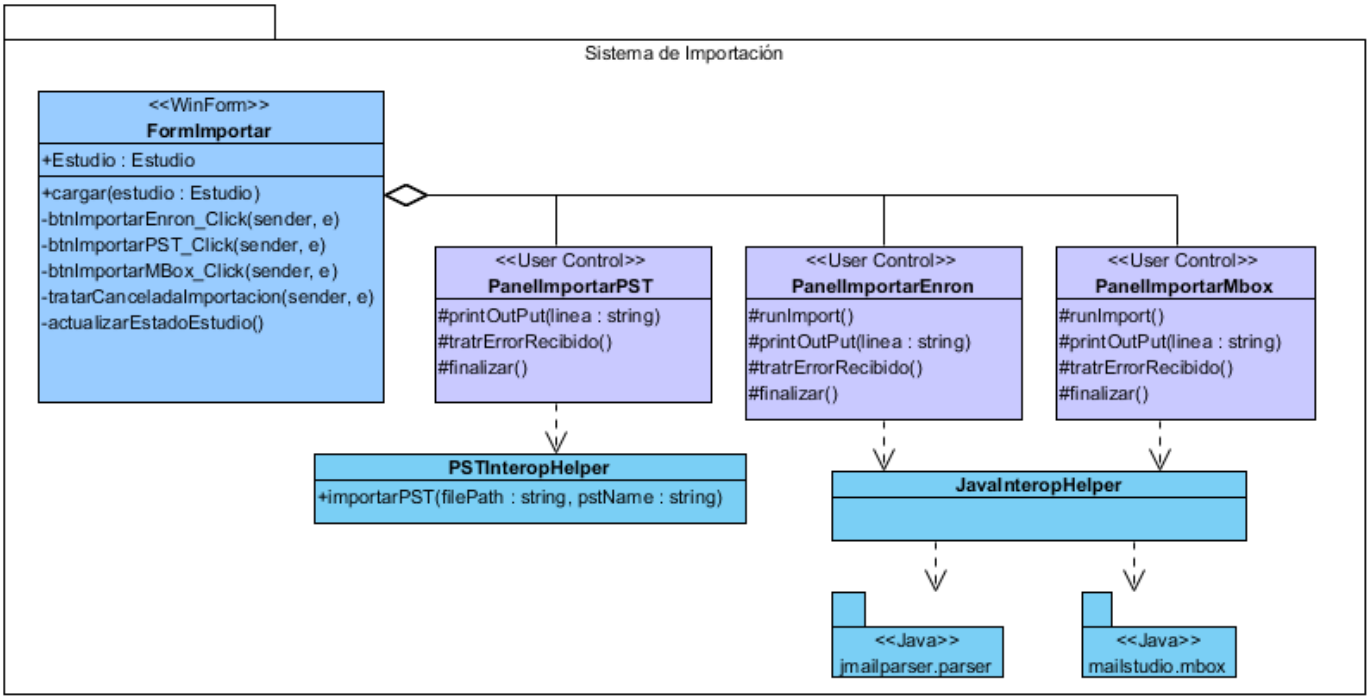


Figura 24: Diseño del sistema de importación.

Se omiten las clases concretas de importación desde ficheros PST. Se utiliza una clase auxiliar *PSTInteropHelper*, que maneja la lógica de la importación a bajo nivel, conectando con la aplicación *Outlook* y accediendo a los datos del repositorio de mensajes de correo electrónico.

La importación de los ficheros planos de *Enron* y ficheros en formato *mbox* se realiza a través de una clase de ayuda *JavaInteropHelper* que utilizará en cada caso una librería *.jar.*

La librería de *jmailparser.parser* ya la disponíamos de un trabajo anterior de importación de correos así que ha parecido preferible reutilizarla más que migrar dicha lógica a plataforma .NET. Por otro lado es interesante disponer de distintas librerías de lecturas de los ficheros fuente de los mensajes de correo electrónico para los casos en que haya que minar corpus de datos con cabeceras y elementos no estándar (como calendarios, recordatorios, etc.).

5.2.2 Diseño de Bases de Datos.

El sistema mantiene una base de datos maestra: *BDEstudio*, y una vez se ha instalado permite tener una base de datos por cada estudio particular que se necesite hacer.

Base de datos de Gestión

La base de datos de Gestión tiene una estructura bastante simple que consta de una única tabla donde se almacenan los datos de conexión de cada base de datos de un estudio concreto:

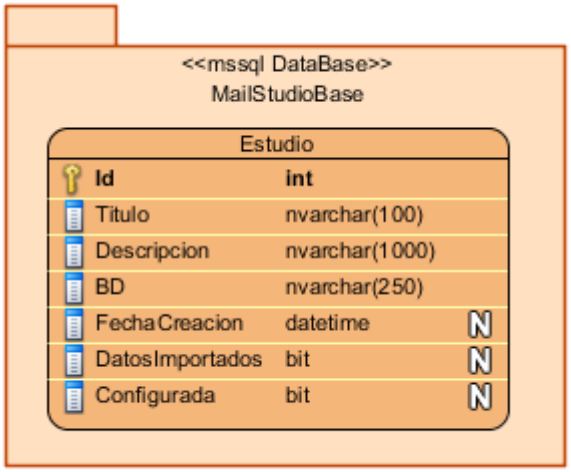


Figura 25: Base de datos MailStudioBase.

Base de datos de un estudio concreto

La base de datos de estudio consta de una tabla en la que se insertan los correos en bruto y una estructura simple para mantener de manera normalizada los destinatarios de cada mensaje de correo electrónico.

La tabla *rawmail* mantiene el hecho de cada correo con los campos más representativos que se han encontrado en el corpus de correo electrónico de Enron.

Asociada a ella tenemos la tabla *rawMailDest* donde se guardan los destinatarios de correo se mantiene el tipo de cabecera (***To***, ***Cc***, ***Bcc***) y los datos de la dirección según la normativa de las RFC.

Las tablas *conf_AgrupacionMail* y *conf_Agrupacion* relacionan las direcciones de destinatarios en diversas agrupaciones que se establecen por dominios de correo electrónico. No obstante el diseño está abierto para definir mediante SQL las agrupaciones bajo otros criterios.

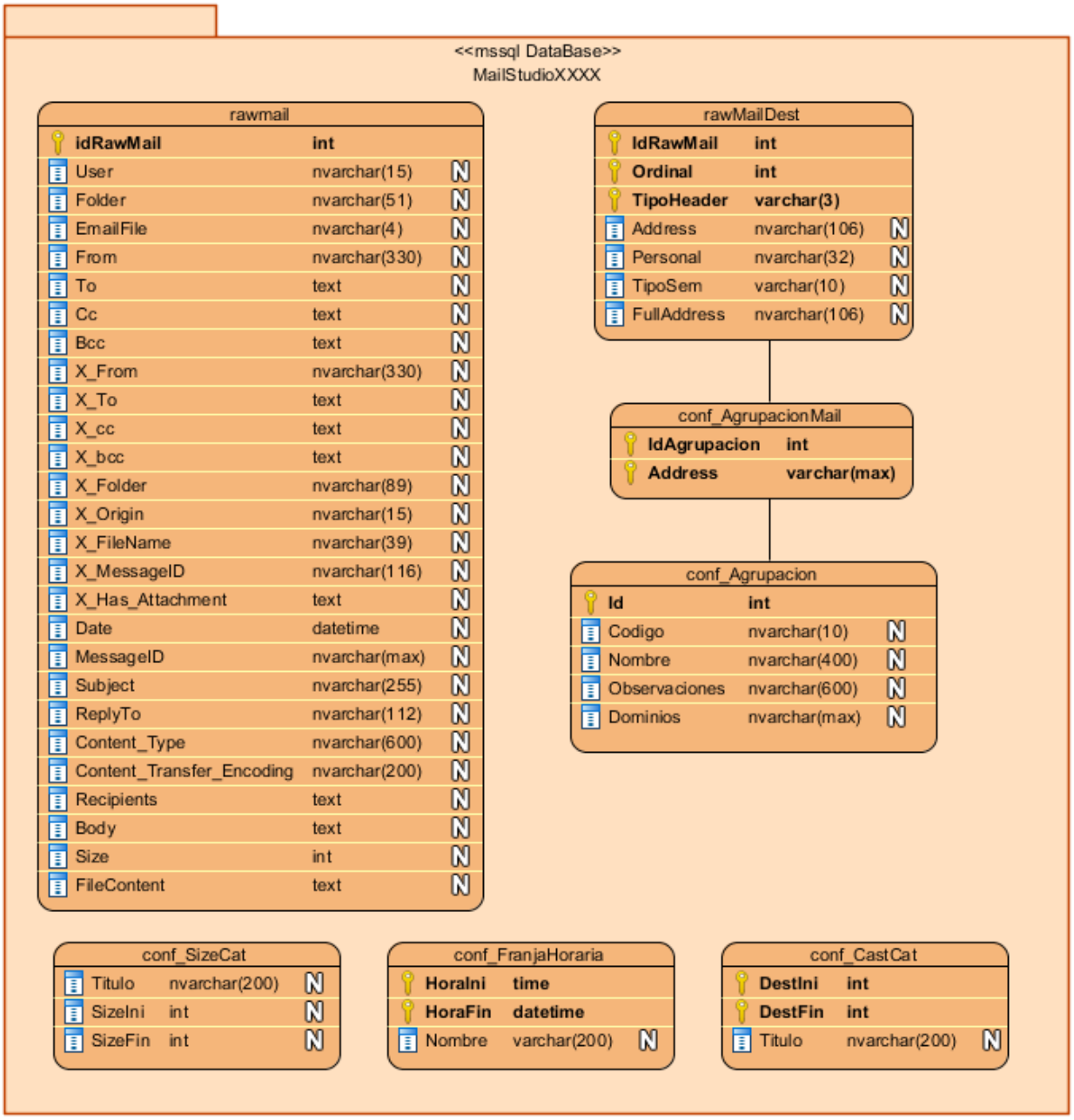


Figura 26: Base de datos de un estudio concreto (1 de 2).

El resto de las tablas son de utilidad para realizar cruces `LEFT JOIN` con dimensiones temporales diversas como días, trimestres o años. Todo el acceso de edición y consulta se hace a través de procedimientos almacenados en la propia base de datos.

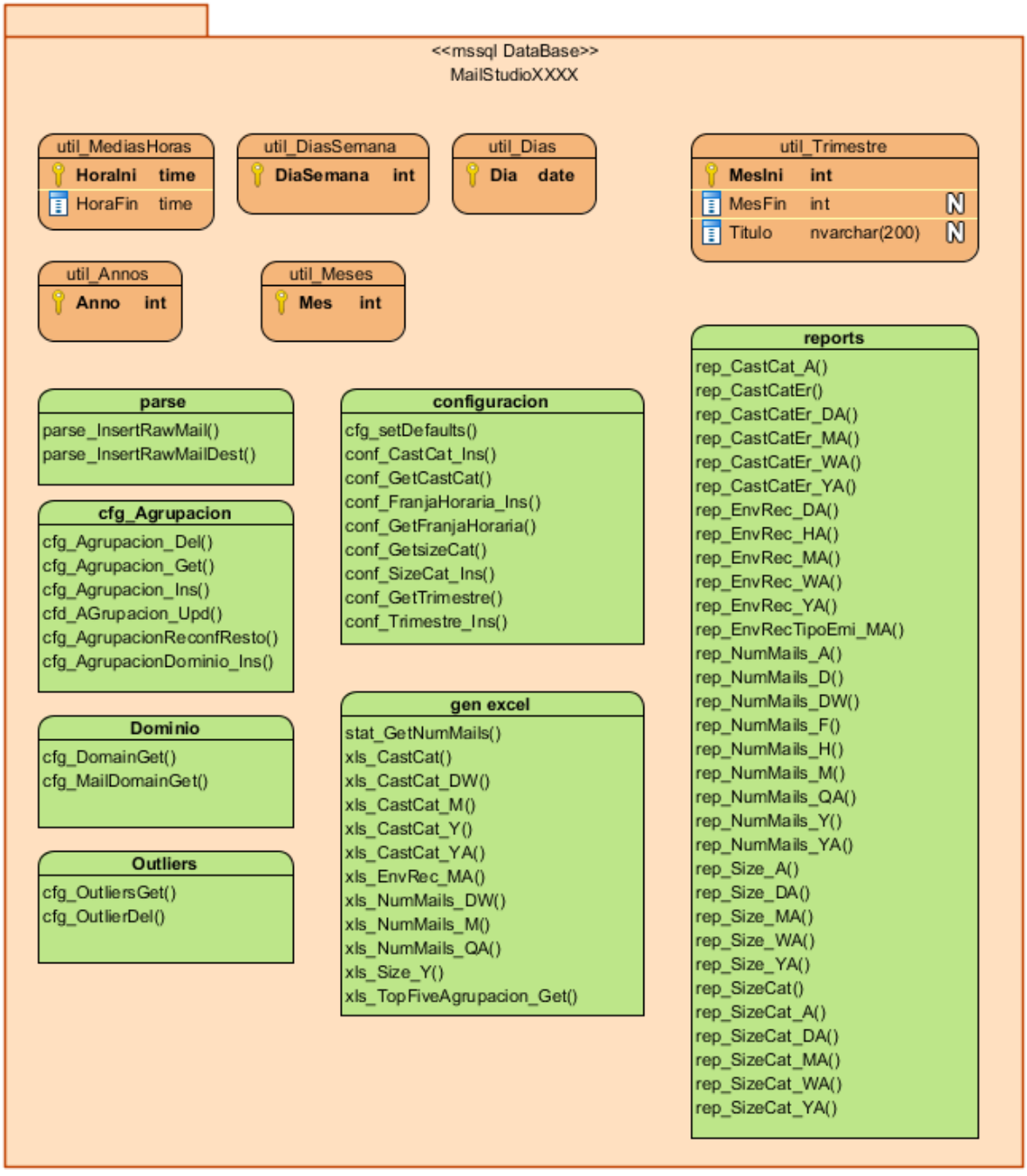


Figura 27: Base de datos de un estudio concreto (2 de 2).

5.2.3 Diseño del Subsistema de informes

Presentamos un esquema de la parte más relevante de la aplicación de informes. Se omiten los formularios y otros componentes como ficheros *rdlc* y *dataset* ADO.NET que se tratarán en el capítulo de **[Detalles de Implementación]**.

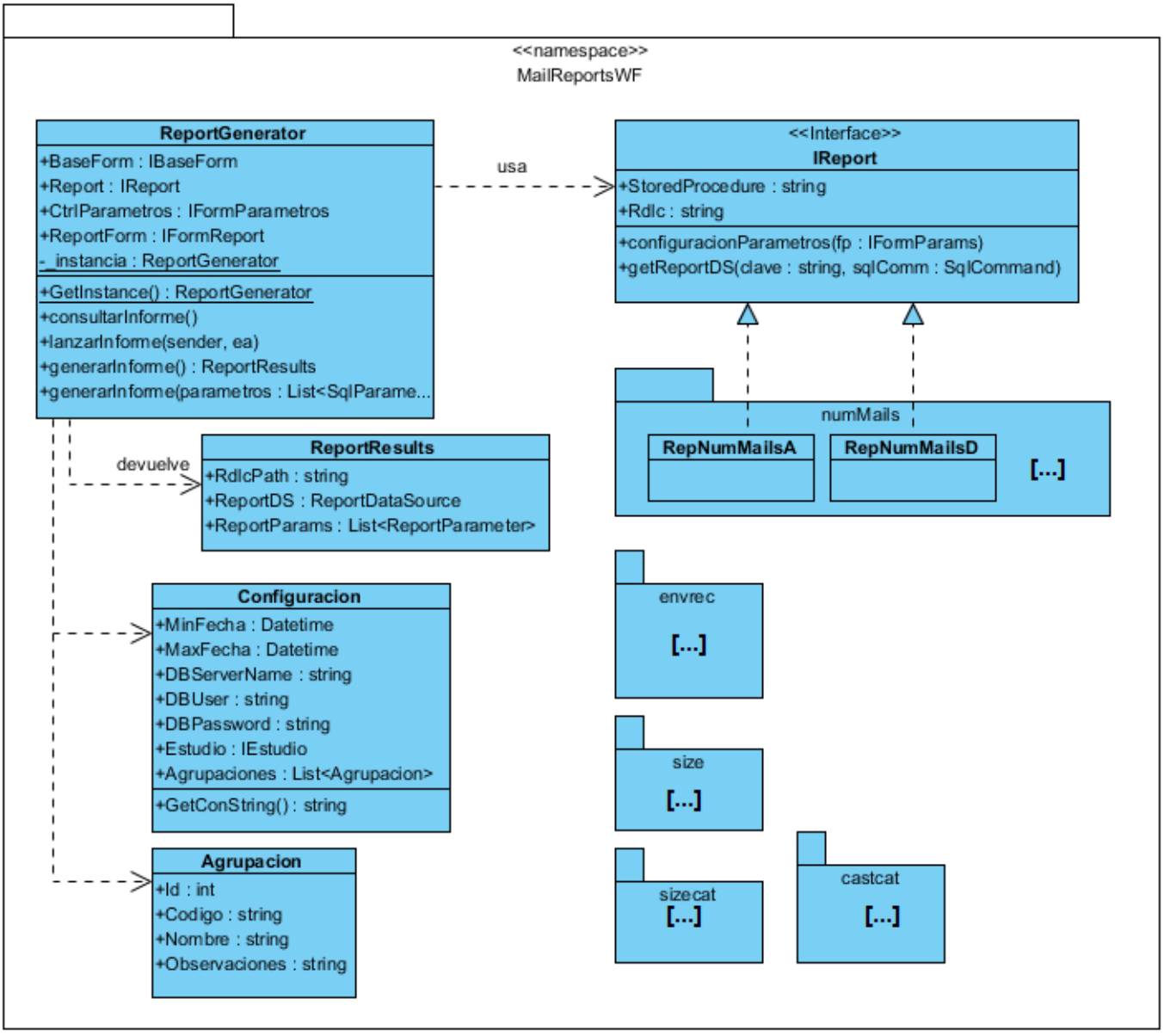


Figura 28: Diseño de la aplicación de informes.

El diseño de la aplicación de informes parte de una clase principal *ReportGenerator* que ataca a clases de informe concretas que contienen las referencias a los ficheros *rdlc*, y de definición de datos de cada informe, todas ellas implementarán la interfaz *IReport*.

Para facilitar la interoperación con los formularios de presentación los resultados de cada consulta se devuelven encapsulados en instancias de `ReportResults`.

La clase de Configuración proporciona los parámetros necesarios para que el generador de informes pueda realizar su labor. La clase de `Agrupacion` se utiliza en diversas partes de esta aplicación para presentar los resultados.

El sistema permite generar un Excel de resumen general para que el usuario pueda tener una visión panorámica de la información presente en los datos.

6 DETALLES DE IMPLEMENTACIÓN

Se especifica la implementación penetrando en algunas cuestiones de carácter especializado que se han considerado remarcables durante la construcción de la aplicación.

Se presenta la parte de la API de *Java Mail* que ha sido empleada para importar distintos formatos de correo electrónico. Se presenta también el modo de incluir funciones de ensamblados *.NET (CLR)* como funciones en la base de datos, aplicado al útil ejemplo de funciones para tratar expresiones regulares.

Por último se profundiza en algunos de los rudimentos de la generación de informes con *Reporting Services*.

6.1 API de *Java Mail*.

Introducción a la librería

La API de la librería *java mail* es una librería de alto nivel para manejar cualquier componente básico de un sistema de correo electrónico. Es una extensión estándar de java y no forma parte del núcleo JDK o JRE (aunque es una parte estándar de Java *Entreprise Edition JEE*).

La librería sigue el patrón de factoría abstracta, las clases principales como *javax.mail.Message* están declaradas abstractas para que el código cliente no deba preocuparse por los detalles de bajo nivel.

Las implementaciones de los protocolos como IMAP, SMTP se incluyen en el paquete, no documentado, de *com.sun.mail*, otras como NJNTP y Exchange están disponibles en paquetes de terceros y algunas como POP están disponibles desde Oracle y por terceras partes. La intención de que las clases principales sean abstractas es aislar de detalles como estos. Enlazando la librería *jar* de la implementación adecuada se puede acceder al sistema por el protocolo adecuado.

La API provee de mecanismos para conectar a los servidores de correo en vivo, leer mensajes, enviar mensajes etc. Para el tipo de estudios analíticos la dinámica consistirá en leer los correos y cargarlos en algún sistema de almacenamiento como ficheros de grafo o base de datos relacional. Solo una parte de la API puede servir a propósitos analíticos.

Al no formar parte del *jdk* o *jre* se debe descargar, está disponible como el paquete *javax.mail.jar* desde la url: www.oracle.com/technetwork/java/javamail/, la versión más moderna es la 1.5.

Librerías	Protocolos	URL
JavaMail	SMTP, IMPA, POP3, Gmail	www.oracle.com/technetwork/java/java
J-Integra Exchange	*Microsoft Exchange*, (DCOM)	j-integra.intrinsyc.com/exchange.asp
exJello	*Microsoft Exchange*, (WebDAV)	www.exjello.org/
ICE MH JavaMail Provider	MH	www.trustice.com/java/icemh
POPpers	POP3	www2s.biglobe.ne.jp/~dat/java/project/poppers
JDAVMail	*Hotmail*, (*WebDAV*)	jdavmail.sourceforge.net
GNU JavaMail	POP3, NNTP, SMTP, IMAP, *mbox*, *maildir*	www.gnu.org/software/classpathx/javamail/
mbox Store	*mbox*	java.net/projects/javamail/pages/MboxStore

Tabla 17: Librerías integrables en javax.mail.

Clases e Interfaces de la librería

La API completa puede consultarse en: https://javamail.java.net/nonav/docs/api/ con el habitual formato de *javadoc*.

Dentro de la jerarquía de clases las más relevantes son la clase abstracta `Message`, que define una enumeración de los tipos de destinatario (***To***, ***Cc***, ***Bcc***), implementa algunos métodos como `getFolder()`, `getReplyTo()` y `getAllRecipients()`. Para trabajar con los mensajes deberemos usar la clase `MimeMessage`, o alguna de las específicas de proveedor como *GmailMessage*.

Para leer un mensaje de un fichero de texto se debe inicializar la sesión y añadir algunas propiedades, una vez hecho basta con pasarle un *InputStream* al constructor de *MimeMessage*:

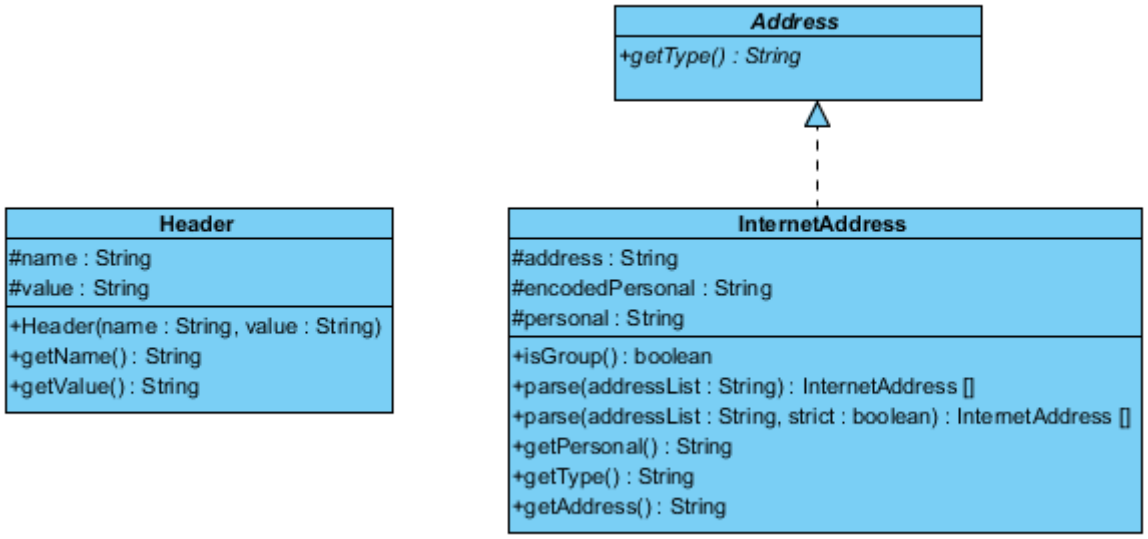
```
Properties props = System.getProperties();
props.put("mail.host", "smtp.dummydomain.com");
props.put("mail.transport.protocol", "smtp");

Session mailSession = Session.getDefaultInstance(props, null);
InputStream source = new FileInputStream(emlFile);
MimeMessage message = new MimeMessage(mailSession, source);
```

Hay que recordar que *JavaMail* como otras librerías está orientada a conectarse a buzones online y utilizar protocolos como SMTP, POP etc… por lo que están muy acopladas con conceptos de conexión como en este caso la `Session`.

Figura 29: Message y MimeMessage en JavaMail 1.5.

Mientras que la clase abstracta `Message` implementa la interfaz `Part`, `MimeMessage`, implementa la interfaz `MimePart` y ofrece un conjunto de métodos suficientemente práctico para poder importar los correos.

Un punto especialmente positivo de utilizar una api como esta es que podrá pasear algunos casos no triviales de direcciones vistos en [**3.2.4 Especificación de Direcciones**] como el de un grupo de direcciones o la inclusión de un nombre antes de la dirección.



Para las direcciones de correo electrónico existe una clase abstracta `Address`, la clase `InternetAddress` proporciona un modo sencillo para poder interpretar una cadena de direcciones.

6.2 Funciones CLR para Expresiones Regulares

Se pueden compilar funciones definidas por el usuario a partir de código .NET. Para la realización de este proyecto se hicieron unos análisis de expresiones regulares, donde la capacidad de funciones TSQL era limitada.

Introducción a las Funciones CLR en base de datos

Desde la versión de 2005 de *M.S. SQL SERVER*, el motor de la base de datos hospeda un entorno de ejecución *CLR (.NET Common Language Runtime*). Éste permite mantener y desplegar ensamblados desarrollados en *Visual Studio* como: procedimientos almacenados, *triggers*, funciones definidas de usuario, funciones de agregado de usuarios y tipos definidos por el usuario. Aunque todo el trabajo que pueda hacerse desde código TSQL debería hacerse sin acudir a ensamblados CLR, estos pueden ser de utilidad para extender comportamientos del motor SQL.

Por defecto el entorno de ejecución CLR está desactivado en el servidor y debe ser específicamente activado con un mando SET. No todas las librerías de .NET pueden usarse al incluir ensamblados en la base de datos, el ejemplo más obvio sería el de la librería `System.Windows.Forms`, empleada para diseñar aplicaciones de escritorio.

Los objetos CLR de *SQL Server* deben incluir siempre atributos (elementos de metadatos) que determinan la intención de un ensamblado particular, clase, método propiedad, etc. Los atributos disponibles se corresponden con alguno de los objetos de SQL que podemos desplegar:

- *SqlProcedure*
- *SqlTrigger*
- *SqlFunction*
- *SqlUserDefinedType*
- *SqlUserDefinedAggregate*

Los tipos de datos a emplear deberían ser aquellos definidos en el *namespace* `System.Data.SqlTypes` para que se puedan mapear directamente hacia los tipos de datos internos de *SQL Server: `SqlInt`, `SqlString`, `SqlBytes`, `SqlBinary`, etc.*

En *Visual Studio* tendremos un tipo de proyecto particular para crear ensamblados CLR que podremos integrar en *SQL Server*.

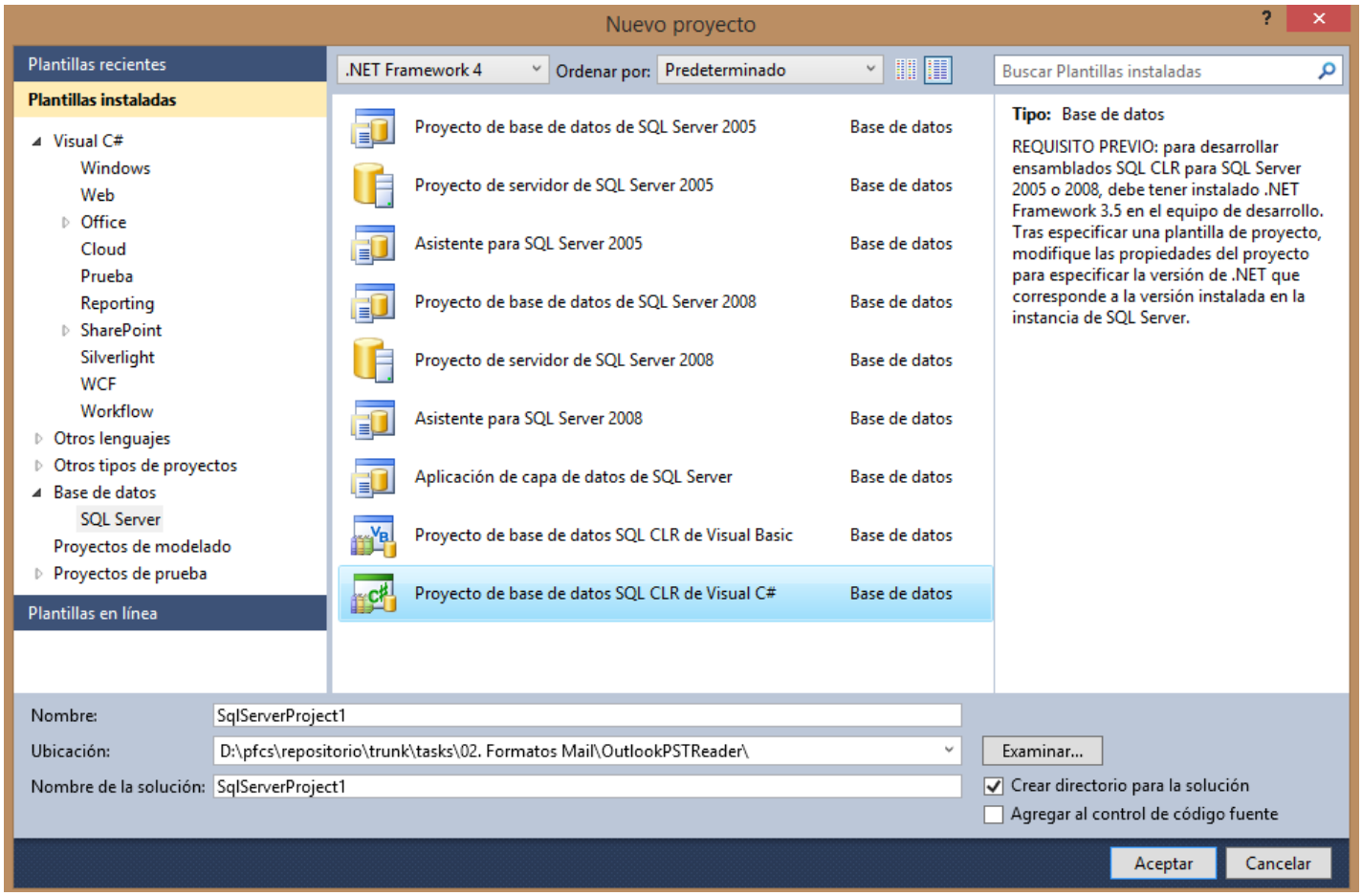


Figura 30: Proyecto de CLR Integrado en SQL Server en Visual Studio 2010.

Desde el IDE de *Visual Studio* podemos agregar los distintos tipos de objetos haciendo *click* derecho sobre el proyecto y haciéndolo luego sobre "*Agregar…*".

Expresiones Regulares TSQL vs CLR

El soporte para expresiones regulares en *SQL Server* se limita a su uso con la función `LIKE`. Esta admite un número reducido de patrones:

- Comodín Múltiple: se indica con el porcentaje '%' y actúa como comodín concordando con cualquier cadena de cualquier extensión. Se utiliza en expresiones del tipo `SELECT * FROM`

`mails WHERE MailUser Like 'Arora%'` que devolvería todos los registros cuyo campo `MailUser` comience con la cadena "`Arora`"

- Comodín Simple: se indica con el carácter del guion bajo: '_' y actúa como comodín de un único carácter. Se podría usar en una expresión del tipo `SELECT * FROM mails WHERE MailUser Like '_rora'` que devolvería todos los registros cuyo campo *MailUser* tenga por cadena '*Arora'*, '*Brora'*, etc.
- Rango de caracteres: se indica de manera muy parecida al uso más común de expresiones regulares, entre corchetes se encierra el rango de caracteres: `[a-Z]` por ejemplo para indicar los caracteres alfanuméricos desde la letra a minúscula hasta la 'Z' mayúscula. La inclusión o no de caracteres con acentos diacríticos, la letra eñe 'ñ', etc, dependen de la opción de *COLLATION* que hallamos configurado en la base de datos.

Por desgracia no se puede especificar las repeticiones, grupos *OR* ni *tokens* opcionales.

El soporte para expresiones regulares desde .NET es mucho más flexible se incluye un lenguaje de expresiones regulares más rico:

- Conjuntos: con la especificación de grupos mediante el par de corchetes "[ ]" se pueden delimitar rangos, `[a-z]` por ejemplo indica al igual que en *SQL* el rango desde la 'a' a la 'z'.
- Clases de Caracteres: por ejemplo `\w` define un *token* palabra; `\d` dígitos decimales; `\s` carácter de espacio, etc.
- Cuantificadores: indican la cardinalidad de un grupo o clase de caracteres: "`*`" indica cero o más, "`+`" indica una o más, "`?`" indica cero o una vez etc.
- Indicadores de Alternancia: La barra "`|`" indica una alternancia entre dos elementos
- Grupos: definen grupos de extracción para obtener como valores la parte de una cadena que coincida contra la expresión regular

Para una referencia completa, se puede consultar la documentación de Microsoft en: https://msdn.microsoft.com/es-es/library/az24scfc%28v=vs.110%29.aspx.

Proceso de despliegue de la librería

Para algunos análisis llevados a cabo en este proyecto fin de carrera, se ha desplegado como funciones de utilidad unas funciones de expresiones regulares de *.NET* en el servidor *SQL Server*. Vamos a enumerar los pasos dados para dicho despliegue.

Existen ejemplos en Internet y en la bibliografía de despliegues similares: **[Nielsen]**, **[Code-Project 19502]**

Es importante recordar que para despliegues en entorno de producción que los ensamblados .NET deberían tener nombres fuertes (*Strong Names*). Si se utiliza *Visual Studio* como se ha indicado antes, y se crea un proyecto de tipo extensión de funcionalidad CLR para *SQL Server*, el proceso será más sencillo por que el ensamblado ya incluirá las referencias adecuadas.

El segundo paso consiste en habilitar la Integración CLR en la base de datos, esto se hace mediante un procedimiento especial:

```
--Enable CLR Integration
exec sp_configure 'clr enabled', 1
GO
RECONFIGURE

GO
```

Una vez hemos compilado y generado el ensamblado con un nombre fuerte debemos importarlo en la base de datos mediante la instrucción SQL:

```
--Create the assembly
CREATE ASSEMBLY [SqlRegEx] FROM 'D:\pfcs\tsql-clr\SqlRegEx.dll'
WITH PERMISSION_SET = SAFE

GO
```

El último paso consiste en crear las funciones de usuario TSQL que harán uso de las funciones .NET que hemos creado y compilado:

```
CREATE FUNCTION [dbo].[ufn_RegExIsMatch] (
      @Input NVARCHAR(MAX),
      @Pattern NVARCHAR(MAX),
      @IgnoreCase BIT)
RETURNS BIT
      AS EXTERNAL NAME SqlRegEx.[SqlClr.SqlRegEx].RegExIsMatch
GO


CREATE FUNCTION [dbo].[ufn_RegExReplace] (
      @Input NVARCHAR(MAX),
      @Pattern NVARCHAR(MAX),
      @Replacement NVARCHAR(MAX),
      @IgnoreCase BIT)
RETURNS NVARCHAR(MAX)
      AS EXTERNAL NAME SqlRegEx.[SqlClr.SqlRegEx].RegExReplace
GO


CREATE FUNCTION [dbo].[ufn_RegExMatches] (
      @Input NVARCHAR(MAX),
      @Pattern NVARCHAR(MAX),
      @IgnoreCase BIT)
RETURNS TABLE (
      Match NVARCHAR(MAX),
      MatchIndex INT,
      MatchLength INT
      )
      AS EXTERNAL NAME SqlRegEx.[SqlClr.SqlRegEx].RegExMatches
GO


CREATE FUNCTION [dbo].[ufn_RegExSplit] (
  @Input NVARCHAR(MAX),
  @Pattern NVARCHAR(MAX),
  @IgnoreCase BIT)
RETURNS TABLE (
      Match NVARCHAR(MAX)
      )
      AS EXTERNAL NAME SqlRegEx.[SqlClr.SqlRegEx].RegExSplit
GO
```

6.3 Desarrollo de informes con *M.S. Reporting Services*

Introducción

Un informe de *Reporting Services* consiste de *Data Sources*, *Data Sets*, los parámetros y la capa de presentación todo ello dentro de uno o más ficheros XML que describen el informe.

El Lenguaje RDL (*Report Definition Language*) es un esquema XML abierto que se usa para representar la recuperación de información y la presentación de la misma en un formato de reporte. Suele incluir elementos para definir conjuntos de datos (*Data Sources*), parámetros, gráficas, tablas y otros elementos de ese tipo necesarios para presentar los datos y formatearlos.

Para la realización de este proyecto se ha optado por utilizar Visual Studio para editar los *rdlcs*.

DataSources

Se puede entender como un origen de datos, contiene la información de una base de datos o un fichero de datos incluyendo el tipo de datos del origen, la cadena de conexión y las credenciales. Todos los informes suelen requerir de un *DataSource*, puede existir uno por reporte (fichero *rdlc*) o tenerlos compartidos para varios reportes dentro de un mismo proyecto.

En nuestro caso hemos diseñado un *DataSource* para cada familia de Informes y nos los hemos llevado a las *Properties* del proyecto:

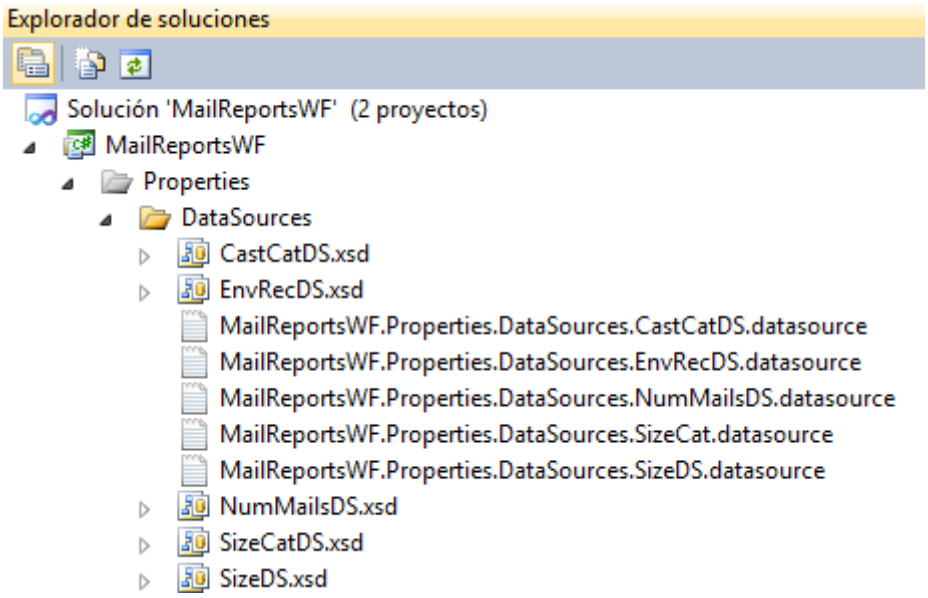


Figura 31: Data Sources del proyecto.

Cada *DataSource* cuenta con un esquema que define las tablas que se pueden consultar (*DataTables*) y los objetos que pueden cargarlas (*DataAdapters*).

Por ejemplo para el *DataSource* de los reportes de tipo *CastCat* tenemos el siguiente esquema:

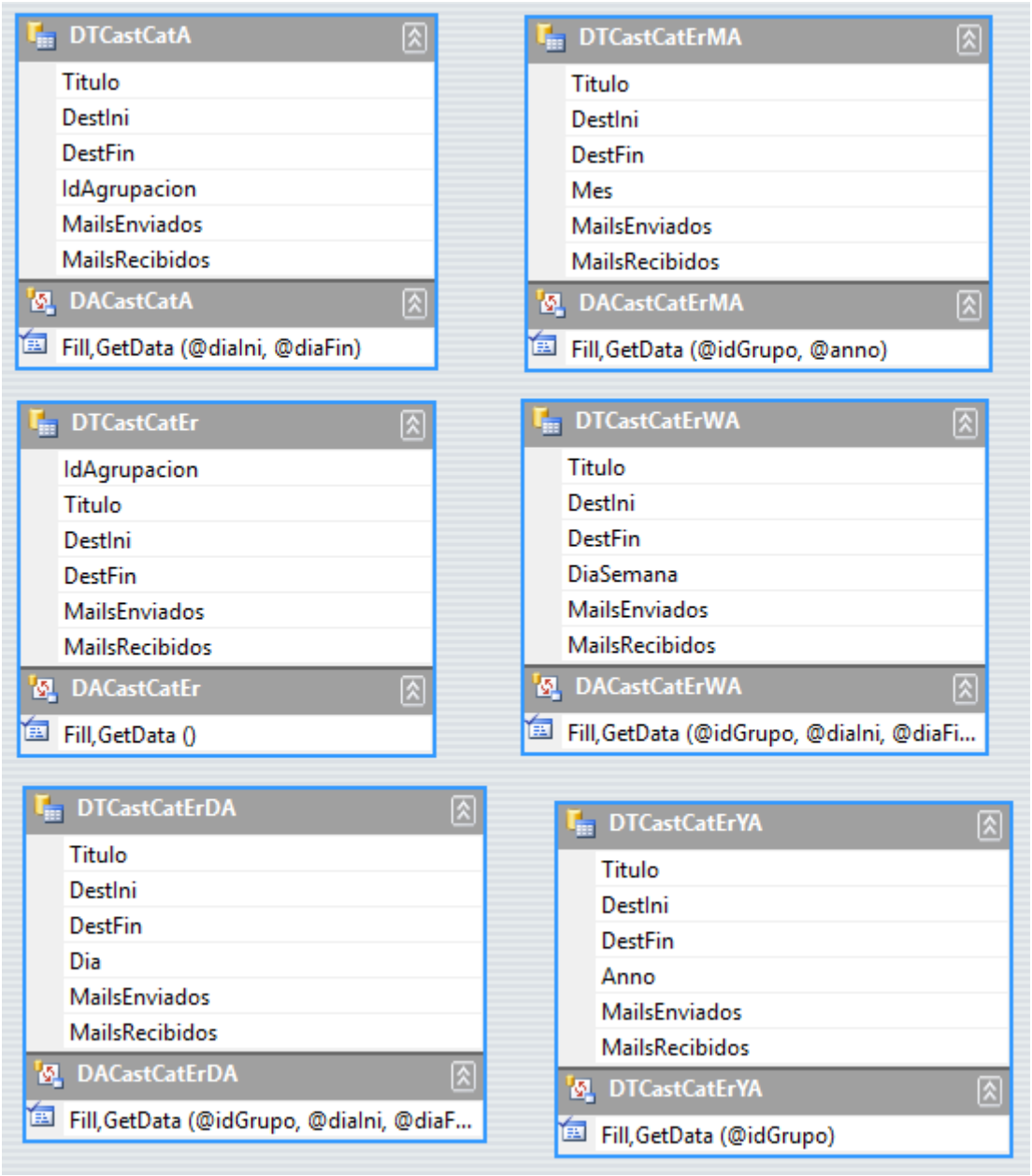


Figura 32: Ejemplo de esquema de datos xsd.

El esquema recoge los conjuntos de resultados con formato como tablas que devuelven cada uno de los procedimientos almacenados de consulta de informes.

Ficheros de definición de informe RDLC

Visual Studio ofrece una interfaz para la edición de ficheros de definición de reportes que facilita mucho el proceso de edición. Se pueden agregar ficheros de informe tanto a proyectos de escritorio como a proyectos de aplicaciones web. Para agregar un nuevo informe podemos utilizar el asistente para informes o comenzar desde cero a editar el fichero *rdlc* haciendo *click* derecho sobre una carpeta o un proyecto en el Explorador de Soluciones y a continuación "*Agregar nuevo informe*":

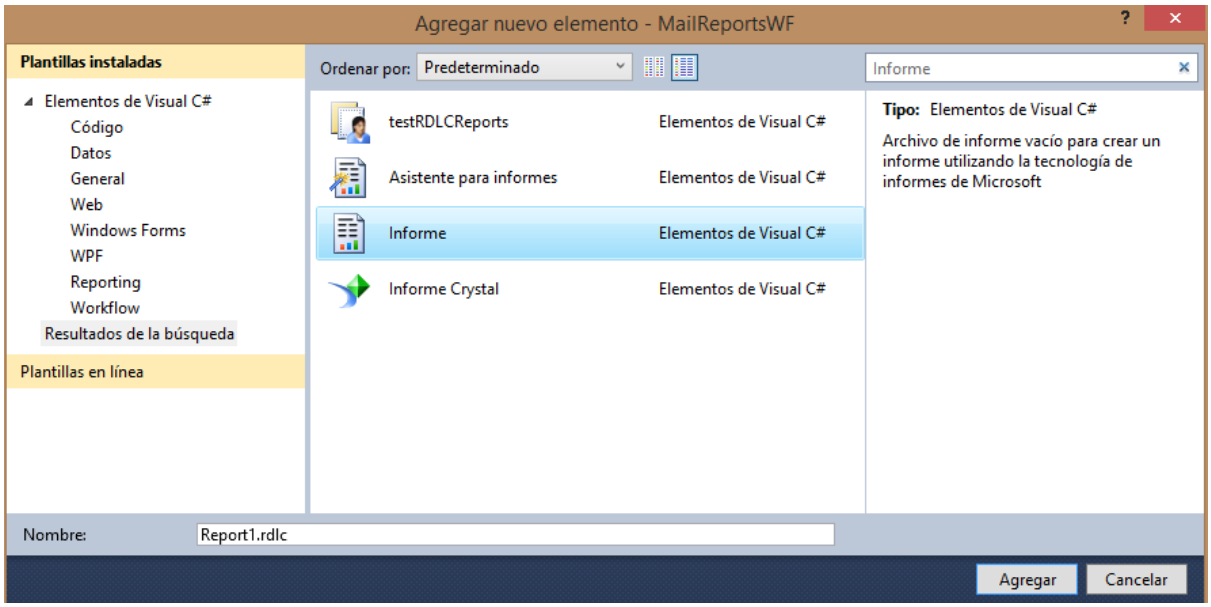


Figura 33: Agregar un nuevo informe en Visual Studio.

En la figura anterior se ha filtrado el tipo de elemento por Informe, para localizarlo más fácilmente se puede acudir a la sección de "*Reporting*" bajo "*Elementos de Visual C#*".

Dentro del editor de informes podremos agregar controles `Tablix` para presentar tablas, listas y matrices de datos, además de cabecera de informe y gráficas.

También se pueden controlar aspectos como los parámetros que recibe el informe: si el informe presenta un número grande de filas, definir el comportamiento al paginar el mismo, establecer una cabecera para todas sus páginas entre otros.

Para una introducción más detallada véase **[Nielsen]**

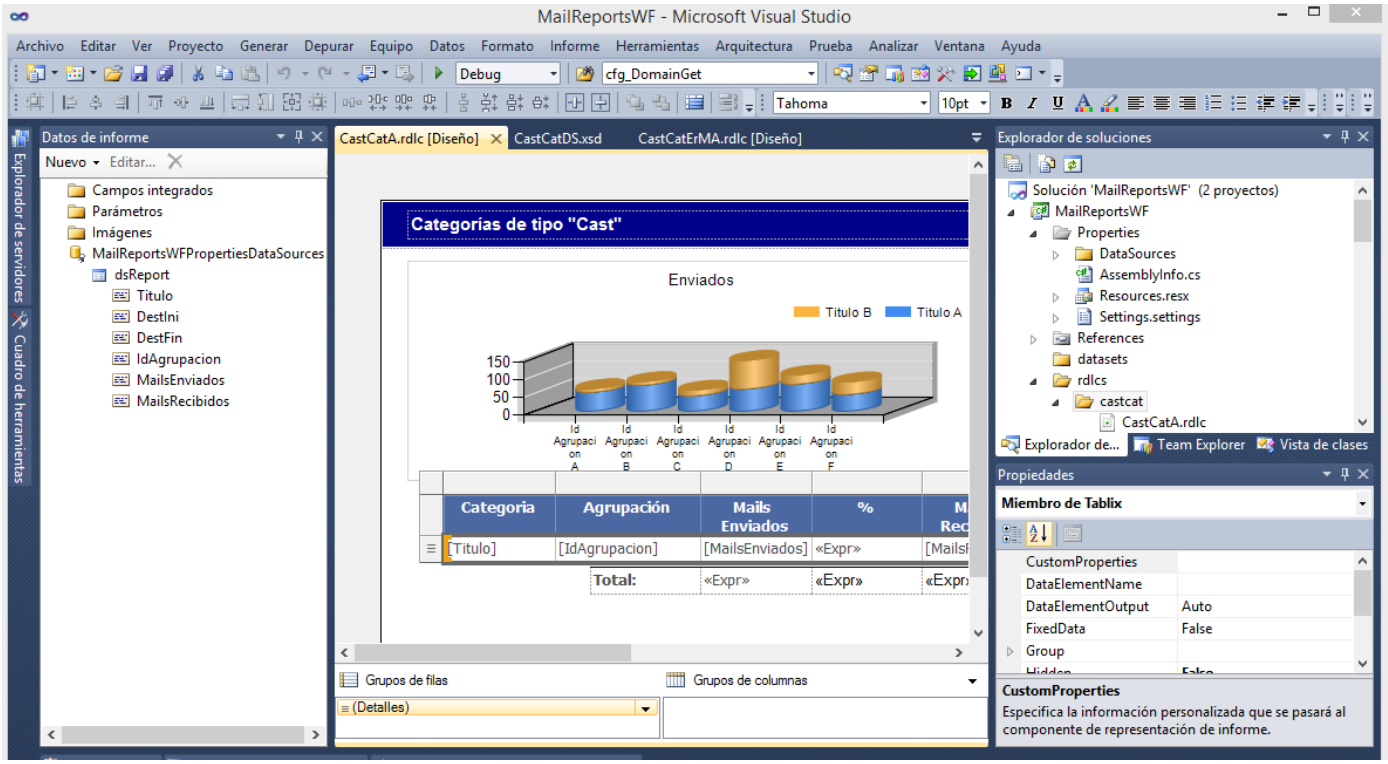


Figura 34: Editor de ficheros RDLC de Visual Studio.

Control Report Viewer

El último elemento de la generación de informes es el control que permite la visualización. Al tratarse de una aplicación de escritorio se ha utilizado un control *ReportViewer* para *WinForms*.

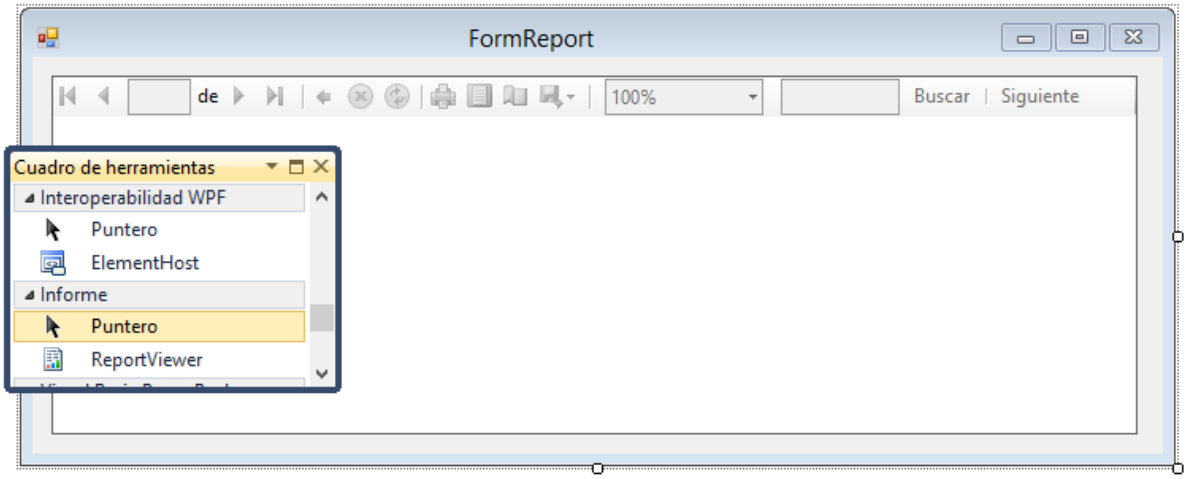


Figura 35: Control Report Viewer.

Al control hay que asignarle uno de los ficheros de definición de informes *rdlc* vistos antes. Aunque la asignación puede ser en tiempo de ejecución lo que nos permite reutilizar el mismo formulario de presentación de informe y el mismo control *ReportViewer* para todos los informes.

Dentro del control hay un abanico de propiedades configurables además de las contenidas en la definición de cada informe.

7 VALIDACIÓN

En este capítulo se detalla un plan de pruebas para verificar la corrección de las funcionalidades que ofrece el sistema, así como la correspondencia de dichas pruebas con los requisitos registrados en el capítulo: **[5.1.3 Requisitos de Usuario]**.

Además se analiza el rendimiento del sistema en distintas operaciones, su extensibilidad para dar cabida a nuevos estudios y análisis.

7.1 Especificación del plan de pruebas

A continuación se listan las pruebas diseñadas para el sistema.

Identificador: TEST-001	Resultado:	**CORRECTO**
Descripción:	Verificar que el sistema se puede instalar en un nuevo entorno *Windows*.	
Metodología:	Probar manualmente que se puede instalar una instancia funcional.	

Tabla 18: Prueba 001.

Identificador: TEST-002	Resultado:	**CORRECTO**
Descripción:	Verificar que se puede crear un nuevo estudio especificando todos sus campos	
Metodología:	Probar manualmente que se pueden crear distintos estudios.	

Tabla 19: Prueba 002.

Identificador: TEST-003	Resultado:	**CORRECTO**
Descripción:	Probar que se puede eliminar un Estudio correctamente y al completo.	
Metodología:	Probar manualmente desde la aplicación que se puede eliminar el estudio y verificar a través de *SQL Management Studio* que la base de datos del estudio se ha eliminado físicamente.	

Tabla 20: Prueba 003.

Identificador: TEST-004	Resultado:	**CORRECTO**
Descripción:	Probar que se puede importar correctamente desde un formato plano el conjunto de datos de Enron.	
Metodología:	Ejecutando la aplicación, sobre un estudio en blanco, probar manualmente que los correos electrónicos se importan correctamente. El proceso se debe poder pausar y en caso de que aparezca un error se debe presentar en el campo de texto de log de la importación. Comprobar al finalizar la importación consultando la base de datos que el número de correos cuadra con el número de correos en el conjunto de datos.	

Tabla 21: Prueba 004.

Identificador: TEST-005	Resultado:	**CORRECTO**
Descripción:	Probar que se importa correctamente desde un formato de datos *mbox*.	
Metodología:	Ejecutando la aplicación sobre un estudio en blanco, importa desde un conjunto de datos *mbox*. Se debe notificar del grado de avance de la tarea, que además debe poderse pausar y reanudar si existe algún error en los datos el sistema debe notificar de los errores y continuar interpretando los siguientes correos.	

Tabla 22: Prueba 005.

Identificador: TEST-006	Resultado:	**CORRECTO**
Descripción:	Probar que se importa correctamente desde un formato PST	
Metodología:	Esta prueba requerirá que el sistema tenga instalado alguna versión de MS Outlook que incluya las librerías de interoperación. Ejecutando la aplicación sobre un estudio en blanco probar a importar un conjunto de datos desde un fichero PST. La aplicación debe mostrar el progreso y los errores aparecidos al leer los correos electrónicos así como poder pausarse y reanudarse la importación.	

Tabla 23: Prueba 006.

Identificador: TEST-007	Resultado:	**CORRECTO**
Descripción:	Probar que el sistema permite definir *Agrupaciones* de direcciones de correo en base al dominio de las mismas.	
Metodología:	Probando manualmente a definir las agrupaciones, borrando las existentes y creando nuevas. Comprobar en paralelo sobre la base de datos que las agrupaciones se reflejan bien en las tablas: `conf_Agrupacion` y `conf_AgrupacionMail`.	

Tabla 24: Prueba 007.

Identificador: TEST-008	Resultado:	**CORRECTO**
Descripción:	Comprobar que los parámetros de configuración básicos se modifican y se leen correctamente en la aplicación. Comprobar además que se eliminan los datos atípicos temporales en función del parámetro de la desviación estándar. Por último comprobar que las agrupaciones de direcciones hechas bajo el criterio del dominio incluyen correctamente las direcciones de correo electrónico.	
Metodología:	Modificar los parámetros de configuración de un *Estudio* y verificar en las tablas correspondientes de la base de datos que se guardan correctamente los valores. Para el caso de los atípicos se puede usar una consulta que devuelva los resultados de fechas más extremos. Haciendo varias pasadas de limpieza de atípicos los valores de las fechas de envío han de reducirse.	

Tabla 25: Prueba 008.

Identificador: TEST-009	Resultado:	**CORRECTO**
Descripción:	Probar las características de limpieza de datos.	
Metodología:	Probar a utilizar diversas configuraciones de parámetros para evitar datos atípicos temporales, contrastar en la base de datos que los mensajes de correo electrónico afectados han sido realmente eliminados.	

Tabla 26: Prueba 009.

Identificador: TEST-010	Resultado:	**CORRECTO**
Descripción:	Probar que el *excel* resumen del corpus se genera correctamente y es legible en *MS Excel* y *Open office*	
Metodología:	Generar el resumen *excel* para varios estudios y abrirlo con las dos aplicaciones.	

Tabla 27: Prueba 010.

Identificador: TEST-011	Resultado:	**CORRECTO**
Descripción:	Probar las variedades de informe de número de correo electrónico	
Metodología:	Contrastar para unos parámetros arbitrarios según el caso que el informe generado se corresponde con los valores que devuelve el procedimiento almacenado. Se debería hacer alguna consulta directa sobre las tablas de datos para contrastar que no hay errores.	

Tabla 28: Prueba 011.

Identificador: TEST-012	Resultado:	**CORRECTO**
Descripción:	Probar las variedades de informe de Enviados vs Recibidos	
Metodología:	Contrastar para unos parámetros arbitrarios según el caso que el informe generado se corresponde con los valores que devuelve el procedimiento almacenado. Se debería hacer alguna consulta directa sobre las tablas de datos para contrastar que no hay errores.	

Tabla 29: Prueba 012.

Identificador: TEST-013	Resultado:	**CORRECTO**
Descripción:	Probar las variedades de informe de "*Carácter Cast*"	
Metodología:	Contrastar para unos parámetros arbitrarios según el caso que el informe generado se corresponde con los valores que devuelve el procedimiento almacenado. Se debería hacer alguna consulta directa sobre las tablas de datos para contrastar que no hay errores.	

Tabla 30: Prueba 013.

Identificador: TEST-014	Resultado:	**CORRECTO**
Descripción:	Probar las variedades de informe de "*Tamaño*"	
Metodología:	Contrastar para unos parámetros arbitrarios según el caso que el informe generado se corresponde con los valores que devuelve el procedimiento almacenado. Se debería hacer alguna consulta directa sobre las tablas de datos para contrastar que no hay errores.	

Tabla 31: Prueba 014.

Identificador: TEST-015	Resultado:	**CORRECTO**
Descripción:	Probar las variedades de informe de "*Categorías de Tamaño*"	
Metodología:	Contrastar para unos parámetros arbitrarios según el caso que el informe generado se corresponde con los valores que devuelve el procedimiento almacenado. Se debería hacer alguna consulta directa sobre las tablas de datos para contrastar que no hay errores.	

Tabla 32: Prueba 015.

Identificador: TEST-016	Resultado:	**CORRECTO**
Descripción:	Verificar que el sistema requiere las librerías de *MS Outlook* para importar ficheros PST.	
Metodología:	Probar manualmente que sin tener las librerías de *MS. Outlook* el sistema avisa al usuario del error de un modo claro.	

Tabla 33: Prueba 016.

Identificador: TEST-017	Resultado:	**CORRECTO**
Descripción:	Verificar que durante el proceso de importación se presenta información de avance de la tarea y la interfaz de usuario no tiene problemas de respuesta frente a los *clicks* del usuario. Probar que la parada y la reanudación de la importación son correctas.	
Metodología:	Probar manualmente la importación para un volumen grande de datos que la interfaz de la aplicación responde y no se queda "*congelada*".	

Tabla 34: Prueba 017.

Identificador: TEST-018	Resultado:	**CORRECTO**
Descripción:	Probar que el sistema restringe el acceso a los informes de un *Estudio* hasta que el usuario ha realizado una importación y una configuración del mismo.	
Metodología:	Probando manualmente a crear varios estudios y a intentar acceder a los informes cuando no realiza importación ni configuración, cuando se realiza importación pero no configuración y cuando si se realizan ambas.	

Tabla 35: Prueba 018.

Identificador: TEST-019	Resultado:	**CORRECTO**
Descripción:	Probar que las gráficas se generan correctamente en el subsistema de informes.	
Metodología:	Generar un informe de cada tipo y comprobar que las gráficas se generan correctamente.	

Tabla 36: Prueba 019.

7.2 Matriz de trazabilidad

Req. vs Pruebas	TEST-001	TEST-002	TEST-003	TEST-004	TEST-005	TEST-006	TEST-007	TEST-008	TEST-009	TEST-010	TEST-011	TEST-012	TEST-013	TEST-014	TEST-015	TEST-016	TEST-017	TEST-018	TEST-019
RUC-001	●																		
RUC-002		●	●																
RUC-003				●	●	●													
RUC-004							●												
RUC-005								●											
RUC-006									●										
RUC-007										●									
RUC-008											●	●	●	●	●				
RUR-001	●																		
RUR-002																●			
RUR-003																	●		
RUR-004																	●		
RUR-005																		●	
RUR-006																			●

Tabla 37: Matriz de trazabilidad Req. vs Pruebas.

7.3 Benchmarking de los procesos

Para la medida de los procesos se ha utilizado una versión SQL Server Express 2008r que limita el funcionamiento a un máximo de 1GB de Memoria RAM, un tamaño máximo de base de datos de 10GB y el uso de un único procesador.

El hardware empleado ha sido un equipo Intel i7-4770S 3.10GHz con 16 GB de memoria RAM.

Se han medido los tiempos de importación para distintos formatos de entrada, y distintos números volúmenes de registros.

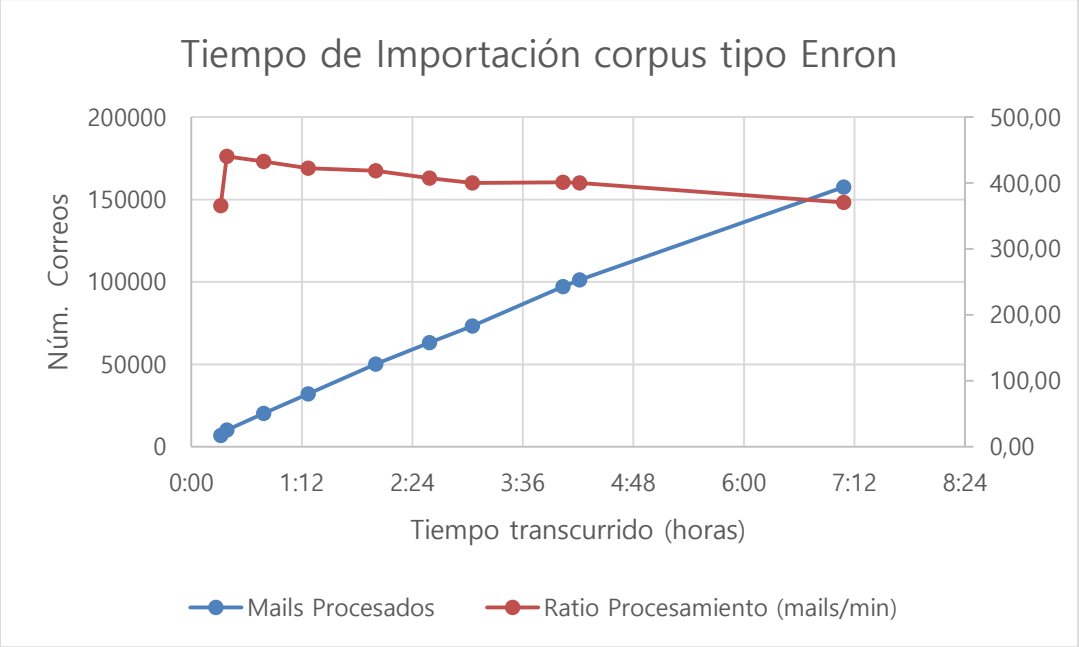


Figura 36: Gráfica de tiempos al importar los datos de Enron.

En el eje x, se ve el tiempo transcurrido en formato de horas y minutos *(hh:mm)*. La serie de correos procesados indica el número de mensajes de correo electrónico que se han procesado mientras que la serie de ratio indica la velocidad de procesamiento en correos por minuto.

Para los formatos PST y *mbox* no se ha dispuesto de un conjunto de datos tan grande como el de Enron, por ello las medidas se reducen a menos de 10.000 correos electrónicos. Para poder tener una equivalencia con respecto a los tiempos y la velocidad de importación respecto al corpus de Enron ofrecemos a continuación la gráfica de tiempos al importar un número menor de correos.

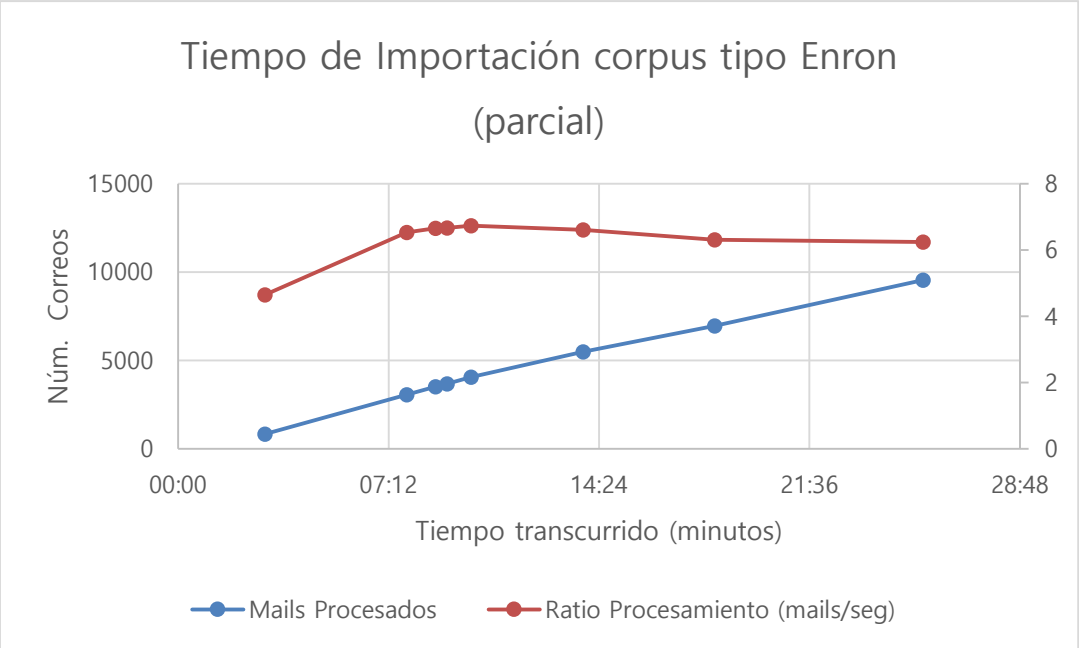


Figura 37: Gráfica de tiempos al importar datos de Enron (parcial).

Para el formato en PST no tenemos un conjunto de datos tan grande como para el caso de Enron. Aquí hemos probado con un fichero PST de correo personal cuyo tamaño roza el *Gigabyte* de tamaño (unos 1000 *MBytes*).

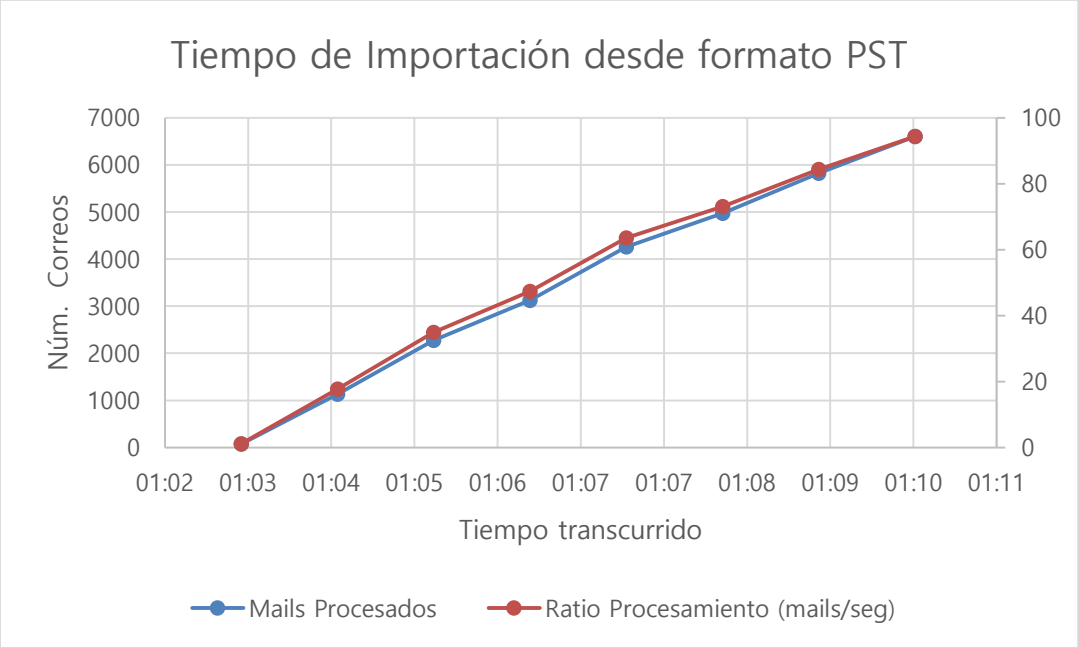


Figura 38: Gráfica de tiempos al importar datos desde un PST.

El caso de la importación desde PST tiene como peculiaridad que se hace un primer barrido de las carpetas de aplicación donde se carga en memoria el contenido de los correos electrónicos. Ese barrido que no supone la importación directa (escritura en base de datos) de ningún correo electrónico consume casi un minuto para el citado conjunto de datos de 1*GB* en la máquina de prueba. Pasado ese tiempo la importación se realiza mucho más rápida que desde cualquiera de los otros formatos donde la lectura y la inserción en *BD* es secuencial. Tal y como se hace en el siguiente caso el conjunto de datos sensiblemente más pequeño al caso de Enron lo que obliga a tomar las medidas en segundos.

Para la importación desde el formato *mbox* tenemos un escenario más similar a la importación desde un formato plano (el conjunto de datos es muy pequeño y por ello se toman las medidas en segundos):

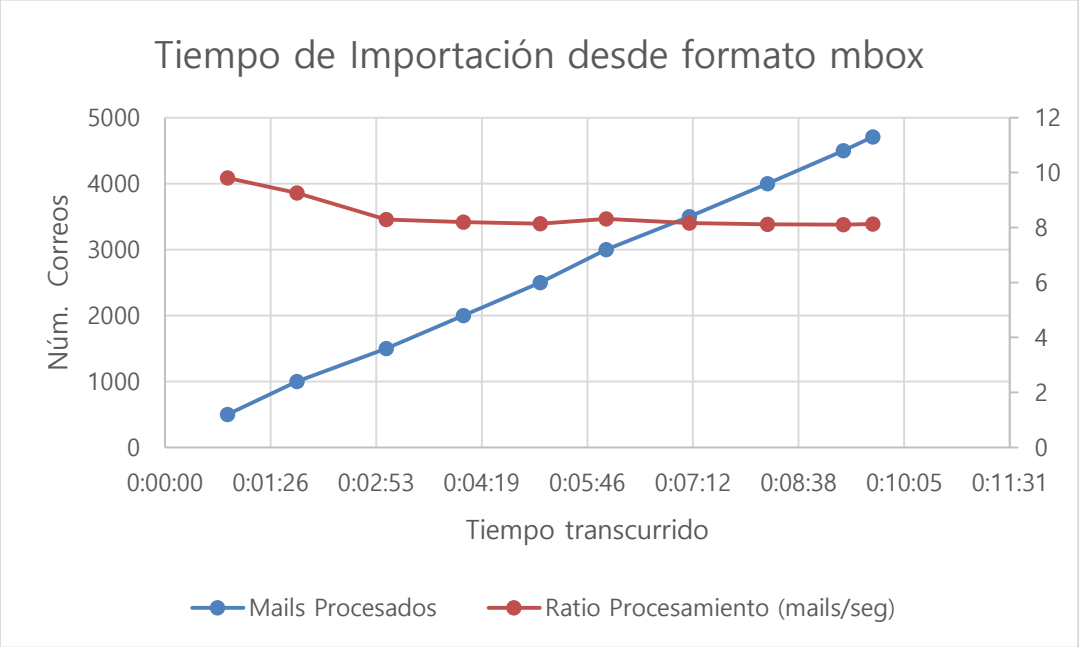


Figura 39: Gráfica de tiempos al importar datos de formato mbox.

Podemos extrapolar los datos de número de correos procesados por segundo a procesados por minuto para comparar la velocidad de importación en función del formato de datos de origen. Las medidas dependen del uso de la máquina y dada la diferencia en los conjuntos de datos no se pueden considerar sino solamente orientativas.

8 CONCLUSIONES Y LÍNEAS FUTURAS

En este capítulo se sacan conclusiones a la luz de los resultados alcanzados y se trazan posibles líneas de estudio futuro. También se aborda un presupuesto para desplegar e implementar el sistema.

También se reflexiona sobre la metodología seguida durante la ejecución del proyecto.

8.1 Conclusiones

Con la ejecución de este proyecto se han alcanzado los objetivos propuestos a su comienzo (véase: **[1.2 Objetivos]**):

- Se ha formalizado un método de análisis. Parte de la estrategia del proceso ha surgido de la experiencia propia al trabajar para importar los datos y realizar diversos análisis sobre ellos y parte ha surgido al revisar el estado de la cuestión donde existen estudios muy buenos que cubren importantes áreas de aplicación de la minería de correo electrónico.
- Se ha desarrollado un sistema de análisis que permite trabajar con varios corpus de estudio o bien tener distintas perspectivas sobre el mismo conjunto de datos. Esto ha probado ser de gran utilidad para tener distintos conjuntos de prueba aun partiendo todos ellos del mismo corpus original. Por ejemplo a la hora de diseñar un algoritmo de expresiones regulares que separara el nombre de la persona de la dirección en direcciones de correo extendidas (***X-From***, ***X-To***…) en el corpus de Enron, poder mantener distintas versiones o alternativas del algoritmo en distintas bases de datos facilita la exploración de soluciones.
- El sistema de reportes utiliza una tecnología de *Microsoft* bien conocida como son los *Reporting Services*, que permite elaborar informes prácticos con bastante rapidez, lo cual puede ser un apoyo a cualquier aplicación relacionada con una minería de correo electrónico en vivo en el contexto de herramientas para analizar rendimiento y problemas de un servidor de correo.
- En esta memoria se tiene como referencia esquemas resumidos de las gramáticas que definen el contenido de un correo electrónico, y resulta de especial utilidad disponer de tales referencias para abordar cualquier estudio o importación de nuevas modalidades de correo electrónico.

8.2 Planificación

Vamos a describir la metodología utilizada en el proyecto así como la organización y planificación de las tareas que se llevan a cabo.

Además se cuantificarán los recursos humanos y materiales necesarios para la realización del proyecto. Todo esto añadido a los costes de componentes tantos físicos (*hardware*) como de *software* conforma un presupuesto que a la fecha de finalización del proyecto reflejan el coste proyectado de llevar a cabo el mismo.

8.2.1 Metodología de trabajo

En un primer momento se presentó la propuesta de proyecto al tutor. Esta fase incluyó la elaboración de un resumen en *Excel* de las consultas que se podrían obtener de un conjunto de correos electrónicos importado en el sistema.

Una vez aceptada la propuesta se ha seguido con una metodología de desarrollo en espiral. Esta metodología se basa en dividir la totalidad del proyecto en varios segmentos de trabajo que o bien se tratan enteros durante una iteración de desarrollo o bien se dividen en más segmentos para su tratamiento. El modelo en espiral combina elementos claves de otras metodologías de trabajo como el modelo en cascada (*waterfall model*) y el prototipado rápido (*rapid prototyping*) en un esfuerzo por mantener las ventajas de una aproximación descendiente (*top-down*) y ascendiente (*bottom-up*). Se pone el énfasis en el análisis iterativo de riesgos de un modo deliberado, que en el caso de este proyecto ha estado en derivar en un análisis demasiado complejo del conjunto de datos que haría el trabajo fuera del alcance de un proyecto fin de carrera por ser demasiado extenso e intrincado.

Los segmentos principales con los que se ha trabajado han sido:

- Análisis Previo
- Importación de datos (Enron)
    - Se comenzó trabajando en la importación de datos de Enron en una maqueta para tratar cuestiones como la lectura de correos electrónicos a través de las librerías de *JavaMail* y el diseño de la base de datos
- Procedimientos de consulta *SQL*
    - El siguiente bloque de trabajo fue el análisis y la elaboración de las tablas auxiliares y procedimientos de consulta que en fases posteriores pudieran servir para presentar los informes estadísticos del corpus.
- Subsistema de Informes
    - En este bloque se diseñó la aplicación de consultas e informes que ya se podía conectar con los procedimientos almacenados de consulta y contaban con datos reales importados del corpus de Enron.
- Sistema de Gestión
    - En este bloque se diseñó la parte de gestión (Altas, Bajas, Modificaciones y Configuración) de Estudios, para poder mantener más de un corpus de correo electrónico en el sistema de un modo estanco.

- Importación de otros formatos (*mbox*, *PST*)
    - En este bloque se implementa la importación de otros formatos agregando al sistema la parte de interfaces gráficas.
- Generación de Excel de Resumen
    - En este bloque se revisó el modelo de documento Excel de resumen que se trató en la propuesta al tutor y se implementó su generación.
- Pruebas de integración y GUI final
    - En este bloque se ha revisado la integración de todos los demás elementos y subsistemas para verificar que el producto final es coherente y consistente con los objetivos.
- Documentación

8.2.2 Presupuesto

Bajo este epígrafe se presenta el desglose del coste de la realización del proyecto. Todos los costes incluyen IVA salvo que se indique lo contrario.

Gastos de Personal

Para realizar el cálculo del presupuesto se han asumido los siguientes supuestos:

- La fecha de inicio será el 22/03/2016
- La fecha de fin será el 15/07/2016
- El periodo suma un total de 80 días laborables contando con 5 días laborables por semana, se excluyen fines de semana y los festivos: 24 y 25 de Marzo, 2 y 16 de Mayo.
- Cada día laborable contabilizará como 5 horas trabajadas.
- Para el proyecto se requerirá de un analista-programador, cuyo coste estimado será de 40,00 €/hora

El desglose de las horas empleadas por actividad del proyecto sería el siguiente:

Actividad	Duración (h)	Recursos	Total
Análisis Previo	60	Analista-Programador	2.400,00 €
Importación de datos (Enron)	40	Analista-Programador	1.600,00 €
Procedimientos de Consulta *SQL*	40	Analista-Programador	1.600,00 €
Subsistema de Informes	60	Analista-Programador	2.400,00 €
Sistema de Gestión	40	Analista-Programador	1.600,00 €
Importación de otros formatos	20	Analista-Programador	800,00 €
Generación de *Excel* resumen	20	Analista-Programador	800,00 €
Pruebas de integración, GUI final.	40	Analista-Programador	1.600,00 €
Documentación	80	Analista-Programador	3.200,00 €
		Total:	**16.000,00€**

Tabla 38: Especificación de actividades y coste.

Costes de Hardware:

Descripción	Coste sin IVA	%Uso dedicado proyecto	Dedicación meses	Periodo de depreciación	Coste imputable
Equipo ASUS…	600 €	100,00%	5	45	66,67 €
				Total:	**66,67 €**

Tabla 39: Costes de Hardware.

Costes de Software:

El coste del software para desarrollo constará de la licencia de *Visual Studio 2010*, una licencia *SQL SERVER 2008 R2 Express Edition* que no tiene coste y una licencia de *Office 365* para seis meses (se toma con algo de margen por si el proyecto sufre retrasos).

Coste de Software de Desarrollo	
Licencia de Software	Coste
Licencia VS 2010	647,00 €
Licencia SQL SERVER 2008 R2 (Edición Express)	0,00 €
Licencia Office 365 (6 meses)	59,94 €
Total:	**706,94 €**

Tabla 40: Costes de Software de desarrollo.

Coste de consumibles:

El coste de productos consumibles para el proyecto como tóner de impresora, hojas en blanco, DVD, etc. se cifra en **200,00**€.

Resumen de Costes

El proyecto ha sido presupuestado en un coste de desarrollo de Treinta y un mil ciento setenta y seis Euros con setenta y ocho Céntimos: ***#31.176,78€#.***

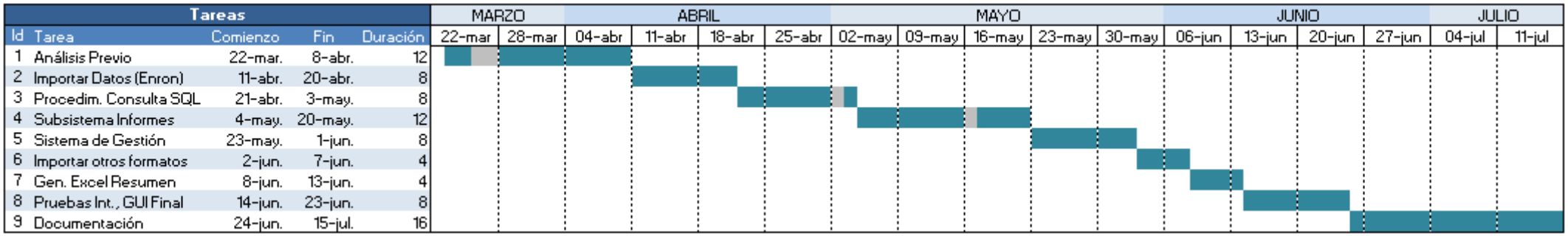
Concepto	Importe
Recursos Humanos	16.000,00 €
HW y Licencias SW	773,61 €
Consumibles	200,00 €
20% Costes indirectos	3.394,72 €
Total presupuesto Inicial:	20.368,33 €
Riesgo (10%)	2.036,83 €
Total antes de Beneficio	22.405,16 €
Beneficio 15%	3.360,77 €
Total sin IVA	25.765,93 €
21% IVA	5.410,85 €
Total con IVA	**31.176,78 €**

Tabla 41: Resumen de Presupuesto.

8.2.3 Planificación

A continuación se presenta el diagrama de *Gantt* con la planificación de proyecto.

En la planificación se reflejan las actividades para completar el sistema y la documentación como la presente memoria. No se incluye la defensa del proyecto dado que es una fecha dependiente de varias agendas: alumno, tutor, examinadores, etc.



Tareas					MARZO		ABRIL				MAYO					JUNIO				JULIO	
Id	Tarea	Comienzo	Fin	Duración	22-mar	28-mar	04-abr	11-abr	18-abr	25-abr	02-may	09-may	16-may	23-may	30-may	06-jun	13-jun	20-jun	27-jun	04-jul	11-jul
1	Análisis Previo	22-mar.	8-abr.	12																	
2	Importar Datos (Enron)	11-abr.	20-abr.	8																	
3	Procedim. Consulta SQL	21-abr.	3-may.	8																	
4	Subsistema Informes	4-may.	20-may.	12																	
5	Sistema de Gestión	23-may.	1-jun.	8																	
6	Importar otros formatos	2-jun.	7-jun.	4																	
7	Gen. Excel Resumen	8-jun.	13-jun.	4																	
8	Pruebas Int., GUI Final	14-jun.	23-jun.	8																	
9	Documentación	24-jun.	15-jul.	16																	

Figura 40: Diagrama de Gantt con la planificación.

8.3 Otras Aplicaciones

Aunque algunas fuentes citadas en el capítulo **[2 Estado de la Cuestión]**, ya hablan sobre ello, mencionaremos además que una red de mensajes de correo electrónico enviados entre varias personas, no es sino una modalidad, la tradicional en las TI, de red social de comunicación. En redes actuales y en los casos de éxito de la última década abundan los mismos tipos de relación: mensaje unipersonal entre dos personas, comunicación de una a varias personas, donde alguna responde a todas, al emisor o a un subconjunto etc.

Muchas magnitudes de estudio como el tiempo de respuesta desde que se recibe un mensaje hasta que se hace algo con él, la frecuencia de mensajes entre un grupo determinados de nodos de la red frente a la frecuencia de mensajes entre el resto de grupos son propias no solo del correo electrónico, que es aquí más herramienta, que finalidad sino del hecho de una comunicación remota y en diferido entre personas.

Para este tipo de redes que llevan emergiendo y propagándose a nivel mundial los métodos de análisis y las herramientas sistemáticas son de gran utilidad no ya en campos como la publicidad inteligente o herramientas de inteligencia artificial para proponer contactos o catalogar mejor los mensajes recibidos, sino también como herramientas de análisis antropológico y de otras en el contexto de las ciencias sociales en un entorno que cada vez desplaza más a la comunicación directa verbal y no verbal entre individuos.

8.4 Líneas Futuras

Las líneas del análisis de redes sociales y el análisis de documentos, siguen en auge y es previsible que mientras las relaciones siguen globalizándose y avanza la tendencia a grafos de usuarios interconectados a escala planetaria, la tecnología evolucione para cumplir requisitos computacionales cada vez más desafiantes.

Bases de datos *NoSQL*

Bases de datos *NoSQL* (*Not Only SQL*) ofrecen oportunidades de implementar sistemas de análisis más potentes en dos de sus variantes más útiles: *Bases de Datos de Documentos* como ***Mongo DB*** o ***Couch DB*** que no siguen un modelo relacional y donde antes que registros se almacenan documentos que además pueden no seguir un esquema rígido.

Todavía más interesantes para análisis estructurales de las redes sociales son las bases de datos de grafos (*Graph Databases*) como ***neo4j*** o ***Apache Giraph*** donde el almacenamiento nativo es una estructura orientada a grafos.

Big Data

Trabajos futuros girarán en torno a las tecnologías emergentes de computación *Big Data*, como *Hadoop* y *Spark*.

Hadoop, (antecesor de *Spark*), es un *framework* de software para montar aplicaciones distribuidas bajo licencia libre. Permite trabajar con miles de nodos y *petabytes* de datos. Se caracteriza por ser una alternativa para manejar volúmenes verdaderamente grandes de datos con modelos de programación simples. Contiene el motor de *MapReduce*, basado en las técnicas de programación *Map* y *Reduce*. Para el almacenamiento se puede utilizar su sistema de ficheros HDFS, bases de datos *Cassandra* para una alta disponibilidad, *HBase*, o Hive más orientada al *Data WareHouse*. Para las tareas relativas a la minería de datos y aprendizaje automatizado cuenta con la librería *Mahout*.

Una opción aún más avanzada sería emplear *Spark*, otro framework que supera la limitación que el proceso *Map-Reduce* impone en cuanto al procesamiento lineal de los datos. Mejora la eficiencia de *Hadoop* en el campo del aprendizaje automatizado aunque su librería *MLib* para la minería de datos no apunta tanta completitud.

Un trabajo futuro podría consistir en implementar un sistema de análisis de corpus en tiempo real con *Spark Streaming*.

Privacidad y ética

A un nivel conceptual el horizonte para el análisis de grandes conjuntos de correo electrónico residirá en el análisis consolidado de varias redes (correo electrónico, *Facebook*, *Twitter*, *Whatsapp*…) de un modo transversal y complementario de tal manera que el universo de la redes sociales, virtuales, se pueda superponer sobre el universo de las relaciones físicas.

Por su naturaleza, es un campo de estudio no exento de cuestiones éticas sobre el derecho a la privacidad. Mientras que el contexto general es el de un número masivo de usuarios que rechazan con ligereza esos derechos en pos de una mayor preeminencia social, la capacidad de esa información para generar valor comercial y estratégico hace recomendable una regulación para la gestión de información de ese carácter.

A pesar de estas cuestiones, este tipo de estudios ofrece grandes oportunidades para estudios que pueden aportar avances beneficiosos a la sociedad.

REFERENCIAS Y BIBLIOGRAFÍA

- **[Bellotti]**: Quality Versus Quantity: E-Mail-Centric Task Management and Its Relation With Overload;
  Victoria Bellotti, Nicolas Ducheneaut, Mark Howard, Ian Smith and Rebecca E. Grinter -Palo Alto Research Center-
  http://students.lti.cs.cmu.edu/11899/files/cp3a-belloti-ducheneaut-howard.pdf

- **[Carn]**: Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters: http://arxiv.org/abs/0810.1355

- **[Code-Project 19502]**: A T-SQL Regular Expression Library for SQL Server 2005
  Steve Abraham
  http://www.codeproject.com/Articles/19502/A-T-SQL-Regular-Expression-Library-for-SQL-Server

- **[Digital Formats PST]**
  Sustainability of Digital Formats Planning for Library of Congress Collections,
  Microsoft Outlook PST 2003 (Unicode)
  http://www.digitalpreservation.gov/formats/fdd/fdd000378.shtml

- **[Enron]**: Enron Email Dataset.
  https://www.cs.cmu.edu/~./enron/

- **[Girvan y Newman]**: Community structure in social and biological networks
  M.Girvan y M.E.J. Newman,
  State University of New Jersey

- **[Gmail Meter]**: Gmail Meter: Advanced Email Analytics & Statistics

http://www.gmailmeter.com

- **[Isasi]**: Redes de Neuronas Artificiales. Un enfoque práctico - Pedro Isasi Viñuela, Ines M. Galvan - Pearson Educación.

- **[Nielsen]**: Microsoft SQL Server 2008 Biblie

- **[Kimball]**: The Data Warehouse Toolkit – Ralph Kimball, Margy Ross, Ed Wilet 3th edition

- **[Kleinberg]**: Hubs, Authorities, and Communities
  Jon. M. Kleinberg
  Cornell University, http://cs.brown.edu/memex/ACM_HypertextTestbed/papers/10.html

- **[Klimt]**: Introducing the Enron Corpus,
  Bryan Klimt, Yiming Yang - Language Technology Inst. –

- **[RFC 821]**: Simple Mail Transfer Protocol
  http://tools.ietf.org/pdf/rfc821

- **[RFC 822]**: Standard For the Format of ARPA Internet Text Messages
  http://tools.ietf.org/pdf/rfc822

- **[RFC 1939]**: Post Office Protocol - Version 3

- **[RFC 1425]**: SMTP Service Extensions
  https://tools.ietf.org/html/rfc1425

- **[RFC 1651]**: SMTP Service Extensions (Julio 1994)
  https://tools.ietf.org/html/rfc1651

- **[RFC 1939]:** Post Office Protocol - Version 3

https://www.ietf.org/**rfc/rfc1939**.txt

- **[RFC 2045]**: Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies
  http://tools.ietf.org/pdf/rfc2045

- **[RFC 2046]**: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types
  https://www.ietf.org/rfc/rfc2046.txt

- **[RFC 2049]**: Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples
  https://www.ietf.org/rfc/rfc2049.txt

- **[RFC 2234]**: Augmented BNF for Syntax Specifications: ABNF
  http://tools.ietf.org/pdf/rfc2234

- **[RFC 2822]**: Internet Message Format
  http://tools.ietf.org/pdf/rfc2822

- **[RFC 5321]**: Simple Mail Transfer Protocol
  https://tools.ietf.org/pdf/rfc5321

- **[RFC 5322]**: Internet Message Format
  http://tools.ietf.org/pdf/rfc5322

- **[RFC 6531]**: SMTP Extension for Internationalized Email
  http://tools.ietf.org/pdf/rfc6531

- **[Send]**: MTA Sendmail,
  http://www.sendmail.com

- **[SendAn]**: Sendmail Analyzer,

http://sareport.darold.net

- **[Shetty]** : The Enron Email Dataset Database Schema and Brief Statistical Report
  Jitesh Shetty, Jafar Adibi

- **[SNAP]**: SNAP, Stanford Network Analysis Project
  STANDFOR UNIVERSITY
  https://snap.stanford.edu

- **[Stevens]**: TCP/IP Illustrated, Volumen 1: The protocols
  Ed: Addison –Wesley, W. Richard Stevens

- **[Tang]**: Email Mining: Tasks, Common Techniques, and Tools
  Guanting Tang, Jian Pei, and Wo-Shun
  Luk School of Computing Science, Simon Fraser University, Burnaby BC, CANADA
  https://www.cs.sfu.ca/~jpei/publications/EmailMining-KAIS.pdf

- **[Trend Q1-2010]**: Threats Trend Report Q1 2010
  http://static.altn.com/Collateral/Security-Threat-Trend-Reports/2010-Q1_Email-Threat-Trend-Report.pdf

- **[Yuan y Harnly]**: Email Thread Reassembly Using Similarity Matching
  Jen-Yuan Yeh y Aaron Harnly

- **[Vadher]**: Email Data Mining: An Aproach to Construct an Organization Position-wise Structure While Performing Email Analysis (2010) Master's Projects. Paper 63.
  San Jose State University, http://scholarworks.sjsu.edu/etd_projects

- **[Vapnik]**: The Nature of Statistical Learning Theory - Vladimir Vapnik -Springer- 1995.